IBM System x3950

Using

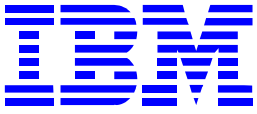
DB2 9 Enterprise Edition

and

SUSE Linux Enterprise Server 10 for Intel EM64T

TPC Benchmark™ C

Full Disclosure Report



First Edition March 26, 2007

Special Notices

The following terms used in this publication are trademarks of **International Business Machines** Corporation in the United States and/or other countries:

IBM System x

IBM

DB2

The following terms used in this publication are trademarks of other companies as follows:

TPC Benchmark, TPC-C, and tpmC are trademarks of the Transaction Processing Performance Council

Microsoft Windows 2003 server and COM+ are registered trademarks of Microsoft Corporation

Linux is a registered trademark of Linus Torvalds

SUSE is a registered trademark of Novell, Inc.

First Edition: March 26, 2007

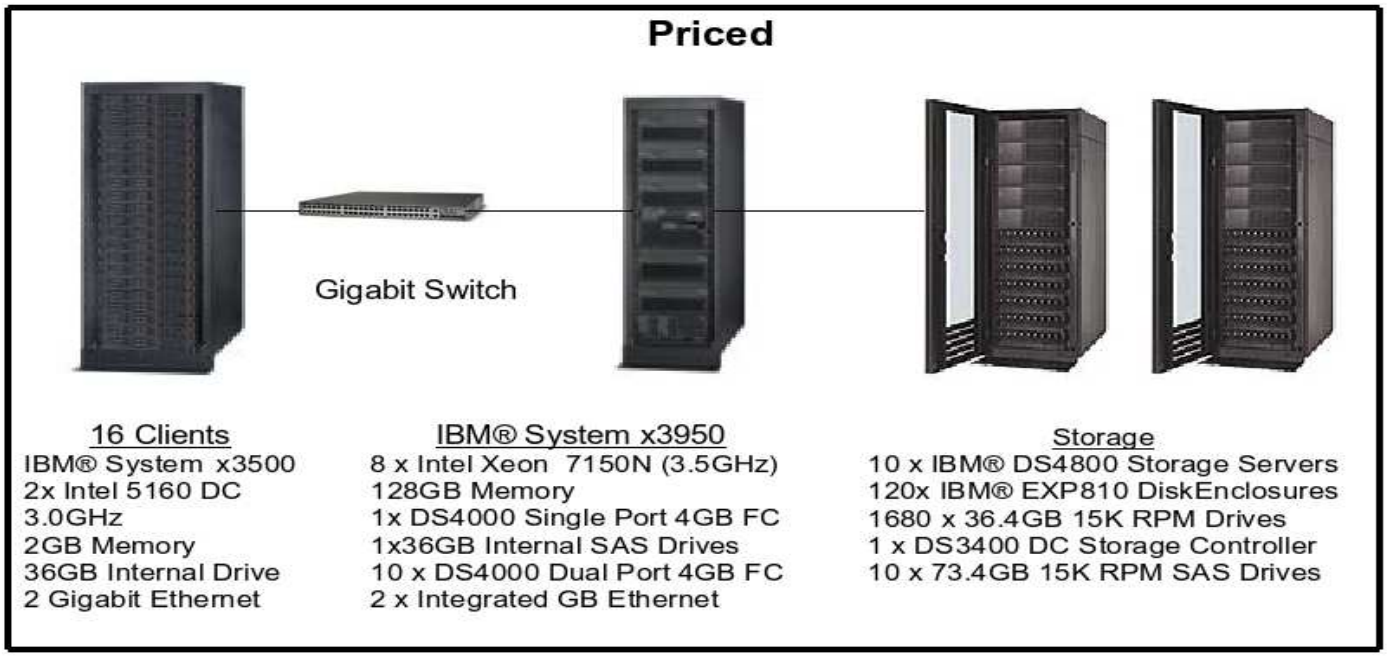
Request for additional copies of this document should be sent to the following address:

TPC Benchmark Administrator
IBM Commercial Performance
Mail Stop 9571
11501 Burnet Road
Austin, TX 78758
FAX Number (512) 838-1852

IBM	**IBM System x3950 DB2® 9**	**TPC-C Rev. 5.8** **Report Date: March 26, 2007**

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$2,784,599 USD	**511,342**	$5.45 USD	**March 26, 2007**

Database Processors/Cores/Threads	Database Manager	Operating System	Other Software	No. Users
8/16/32 3.5 GHz Intel Xeon 7150N	DB2 9	SLES 10	Microsoft Visual C++ Microsoft COM+	416000



System Components	Each of the 16 Clients		Server	
	Quantity	Description	Quantity	Description
Processors/Cores/ Threads	2/4/4	3.0GHz 2MB L2 Xeon 5160 Dual Core	8/16/32	3.5 GHz Intel Xeon 7150N
Memory	4	512 MB	32	4 GB
Disk Controllers	1	SAS	2 10 1 10 1	Integrated SAS 4Gb dual-port FC Adapters 4Gb single-port FC Adapters IBM DS4800 Controllers for Data IBM DS3400 Controller for Log
Disk Drives	1	36 GB	1680 1 10	36.4GB 15K RPM FC 36GB 10K RPM SAS 73.4GB 15K RPM SAS
Total Storage		576 GB		61,152 GB
Terminals	1	System Console	1	System Console

IBM Corporation	IBM System x3950 c/s DB2 9				TPC-C Revision 5.8 Report Date: March 26, 2007		
Description	**Part Number**	**Third Party Brand**	**Pricing**	**Unit Price**	**Quantity**	**Extended Price**	**3-Yr. Maint. Price**
Server Hardware							
IBM System x3950 (2 x Intel Xeon Processor 7150N 3.5GHz 667MHz, 2 x 1 MB L2 Cache and 16MB L3 Cache)	8878-5RU	IBM	1	21,269	1	21,269	
Intel Xeon Processor 7150N 3.5GHz/2x1MB L2, 16MB L3	43V4553	IBM	1	5,699	6	34,194	
x3950 E (0 Processors, 0 Memory, 2 Memory Cards)	8879-1RU	IBM	1	6,999	1	6,999	
Scalability Cable 2.3.M	13M7414	IBM	1	299	2	598	
8GB (2x4GB) PC2-3200 CL3 2RX4 ECC DDR2 SDRAM RDIMM	30R5145	IBM	1	5,569	16	89,104	
Active Memory 4-Slot Memory Expansion Card	41Y5000	IBM	1	499	2	998	
IBM 36.4GB 10K 2.5 inch SAS Drive	40K1051	IBM	1	279	1	279	
IBM T115 15-inch TFT Display	494215U	IBM	1	209	1	209	
IBM Preferred Pro USB Keyboard	40K9584	IBM	1	29	1	29	
IBM 3-Button Optical Mouse - Black - USB	40K9201	IBM	1	19	1	19	
ServicePac for 3-Year 24x7x4 Support (x3950)	96P2688	IBM	1	3,390	1		3,390
ServicePac for 3-Year 24x7x4 Support (x3950 E)	10N3059	IBM	1	1,695	1		1,695
ServicePac for 3-Year 24x7x4 Support (Display)	30L9183	IBM	1	90	1		90
					Subtotal	153,698	5,175
Server Storage							
IBM DS4000 FC 4Gbps - PCI-X Dual Port HBA	39M5895	IBM	1	2,485	10	24,850	
IBM TotalStorage DS4800 Midrange Disk Subsystem	1815-82H	IBM	1	65,245	10	652,450	
IBM Short Wave SFP Module (4 Pack)	22R0483	IBM	1	550	55	30,250	
IBM 1m LC-LC Fibre Channel Cable	39M5696	IBM	1	79	240	18,960	
IBM 5m LC-LC Fibre Channel Cable	39M5697	IBM	1	129	20	2,580	
IBM TotalStorage DS4000 EXP810 Storage Exp. Unit	1812-81H	IBM	1	6,000	120	720,000	
36.4GB 15K 2Gbps FC Drive	06P5772	IBM	1	1,115	1680	1,873,200	
IBM System Storage DS3400 Dual Controller	1726-42X	IBM	1	9,295	1	9,295	
IBM Short Wave SFP Module	19K1271	IBM	1	499	1	499	
73.4GB 15K SAS Hot-Swap HDD	40K1043	IBM	1	349	10	3,490	
IBM DS4000 FC 4Gbps - PCI-X Single-Port HBA	39M5894	IBM	1	1,485	1	1,485	
1M Fibre Optic Cable	39M5696	IBM	1	79	1	79	
IBM S2 42U Standard Rack	93074RX	IBM	1	1,489	10	14,890	
ServicePac for 3-Year 24x7x4 Support (EXP810)	41L2768	IBM	1	760	120		91,200
ServicePac for 3-Year 24x7x4 Support (DS4800)	96P2062	IBM	1	1,087	10		10,870
ServicePac for 3-Year 24x7x4 Support (DS3400)	41L2768	IBM	1	760	1		760
ServicePac for 3-Year 24x7x4 Support (Rack)	41L2760	IBM	1	300	10		3,000
					Subtotal	3,352,028	105,830
Server Software							
DB2 ESE 9		IBM	2	270.59	800	216,472	
Extended Systems - SW License and Maintenance 12 Months							
SW Maintenance Renewal - 1 Year		IBM	2	12.89	1600		20,624
SUSE Linux Enterprise Server 10 for x86, AMD64, and							
Intel EM64T with Priority Support for 3 Years	4584142100	SUSE	3	3,748	1	3,748	
					Subtotal	220,220	20,624
Client Hardware							
Linksys ProConnect KVM Switch - 2 ports (2 spares)	430446		6	40	3	120	
IBM System x3500 (Dual-Core Intel Xeon 5160 3.00GHz, 2x512MB RAM	7977-92U	IBM	1	3,399	16	54,384	
3.00GHz/1333MHz/2x2MB L2 Xeon Processor 5160 Upgrade	40K1236	IBM	1	1,575	16	25,200	
1GB (2x512MB) PC2-5300 CL5 ECC DDR2 FBD 667MHz	39M5782	IBM	1	459	16	7,344	
36.4GB 15K SAS Hot-Swap Drive	40K1042	IBM	1	249	16	3,984	
ServicePac for 3-Year 24x7x4 Support (x3550)	96P2250	IBM	1	586	16		9,376
					Subtotal	91,032	9,376
Client Software							
Microsoft Windows Server 2003 Web Edition with COM+	484143	Microsoft	6	296	16	4,736	
Microsoft Visual C++ Professional 6.0	254-00170	Microsoft	4	109	1	109	
Microsoft Problem Resolution Services		Microsoft	4	245	1		245
					Subtotal	4,845	245
Network Components							
D-LINK DGS-1024D 24-Port 10/100/1000 Switch (2 spares)	DGS1024D		5	230	3	690	
Ethernet Cable (2 spares)	CC5E-B14B		7	3	20	60	
					Subtotal	750	
					Total	3,822,573	141,250
Large Purchase Discount (See Note 1.)	31.72%		1			1,179,224	

Pricing: 1 - IBM - 1-888-SHOP-IBM, ext. 5821; 2 - IBM; 3 - Novell; 4 - Microsoft;
5 - compuplus.com; 6 - CDW.com; 7 - newegg.com

Note 1: Discount based on IBM Direct guidance applies to all line items where Pricing=1. Pricing is for this system or one of similar size.

Note 2: Pricing for DB2 9 is based on Value Units (VUs) as shown in the price quote in Appendix D.

Note 3: Fifteen x335s were substituted for the priced clients.

Audited by Francois Raab, InfoSizing, Inc. (www.sizing.com)

Three-Year Cost of Ownership USD:	$2,784,599
tpmC:	511,342
$ USD/tpmC:	$5.45

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that stated prices are not available according to these terms, please inform the TPC at pricing @ tpc.org.

Numerical Quantities Summary for the IBM System x3950

MQTH, computed Maximum Qualified Throughput: 511,342 tpmC

Response Times (in seconds)	90th %	Average	Maximum
New Order	1.34	0.70	11.29
Payment	1.36	0.68	11.15
Order-Status	1.14	0.65	10.97
Delivery (interactive)	0.48	0.21	9.89
Delivery (deferred)	0.38	0.22	1.19
Stock-Level	1.76	1.16	12.00
Menu	0.63	0.23	9.89

Response time delay added for emulated components was 0.1 seconds

Transaction Mix, in percent of total transactions	Percent
New Order	44.96%
Payment	43.02%
Order-Status	4.01%
Delivery	4.01%
Stock-Level	4.01%

Keying/Think Times (in seconds)	Min.	Average	Max.
New Order	18.00/0.00	18.00/12.04	18.02/120.31
Payment	3.00/0.00	3.00/12.04	3.02/120.33
Order-Status	2.00/0.00	2.00/10.04	2.02/100.31
Delivery	2.00/0.00	2.00/5.04	2.02/50.31
Stock-Level	2.00/0.00	2.00/5.04	2.02/50.31

Test Duration

Ramp-up Time	1 hour 26 minutes
Measurement interval	2 hours 00 minutes
Transactions during measurement interval (all types)	136,488,049

Checkpoints

Number of checkpoints	N/A
Checkpoint interval	N/A

Table of Content

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.8 dated December, 2006, for measurements on the IBM System x3950. The software used on the IBM System x3950 includes SUSE Linux Enterprise Server 10 for Intel EM64T operating system and DB2 9 data server. Microsoft COM+ is used as the transaction manager.

IBM System x3950

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM System x3950	DB2 9	SUSE Linux Enterprise Server 10 for Intel EM64T

Total System Cost	TPC-C Throughput	Price/Performance
• Hardware • Software • 3 Years Maintenance	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC
$2,784,599 USD	511,342	$5.45 USD

Preface

TPC Benchmark™ C Standard Specification was developed by the Transaction Processing Performance Council (TPC). It was released on August 13, 1992 and updated with revision 5.8 in December 2006.

This is the full disclosure report for benchmark testing of the IBM System x3950 and DB2 9 according to the TPC Benchmark™ C Standard Specification.

TPC Benchmark™ C exercises the system components necessary to perform tasks associated with that class of on-line transaction processing (OLTP) environments emphasizing a mixture of read-only and update intensive transactions. This is a complex OLTP application environment exercising a breadth of system components associated by such environments characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Data bases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention on data access and update

This benchmark defines four on-line transactions and one deferred transaction, intended to emulate functions that are common to many OLTP applications. However, this benchmark does not reflect the entire range of OLTP requirements. The extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

Benchmark results are highly dependent upon workload, specific application requirements, and systems design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-C should not be used as a substitute for a specific customer application benchmarks when critical capacity planning and/or product evaluation decisions are contemplated.

The performance metric reported by TPC-C is a “business throughput” measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

0 General Items

0.1. Application Code Disclosure

The application program (as defined in Clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input and output functions.

Appendix A contains the application code for the five TPC Benchmark™ C transactions.

0.2. Benchmark Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

This benchmark was sponsored by **International Business Machines Corporation.**

0.3. Parameter Settings

Settings must be provided for all customer-tunable parameters and options which have been changed from the defaults found in actual products, including but not limited to:

- *Data Base tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters.*

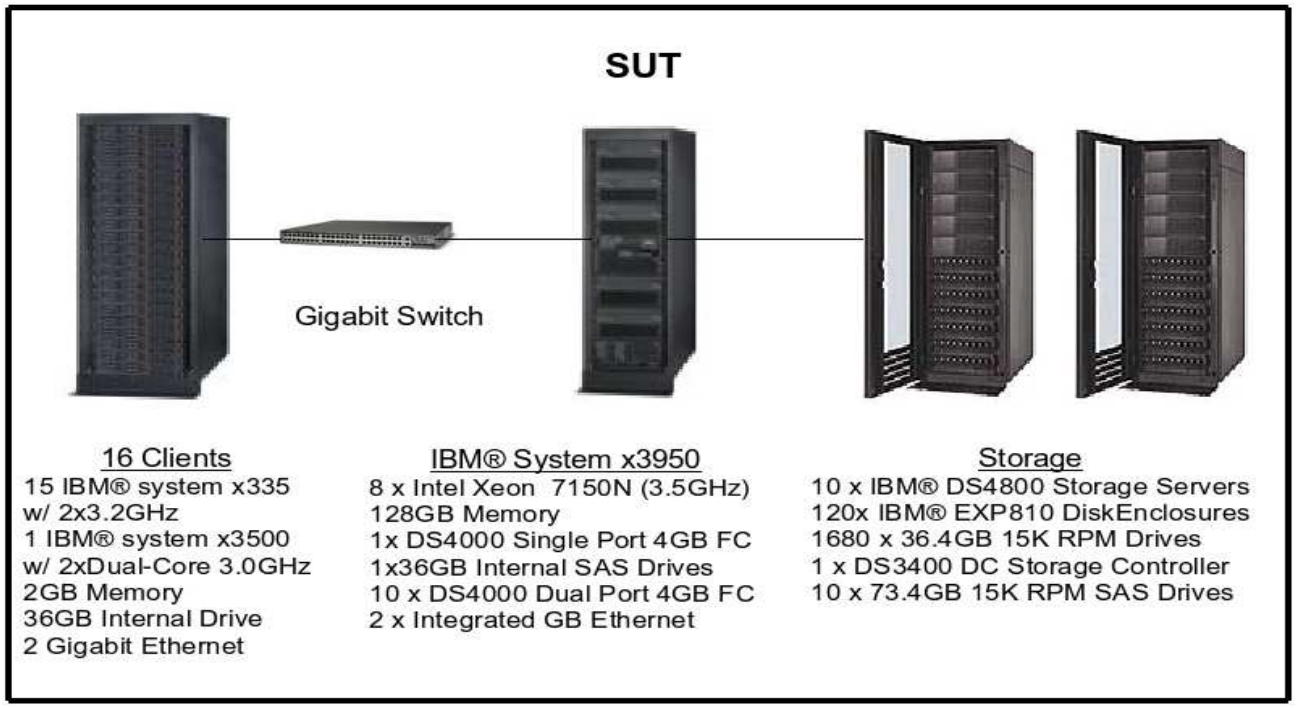
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

0.4. Configuration Diagrams

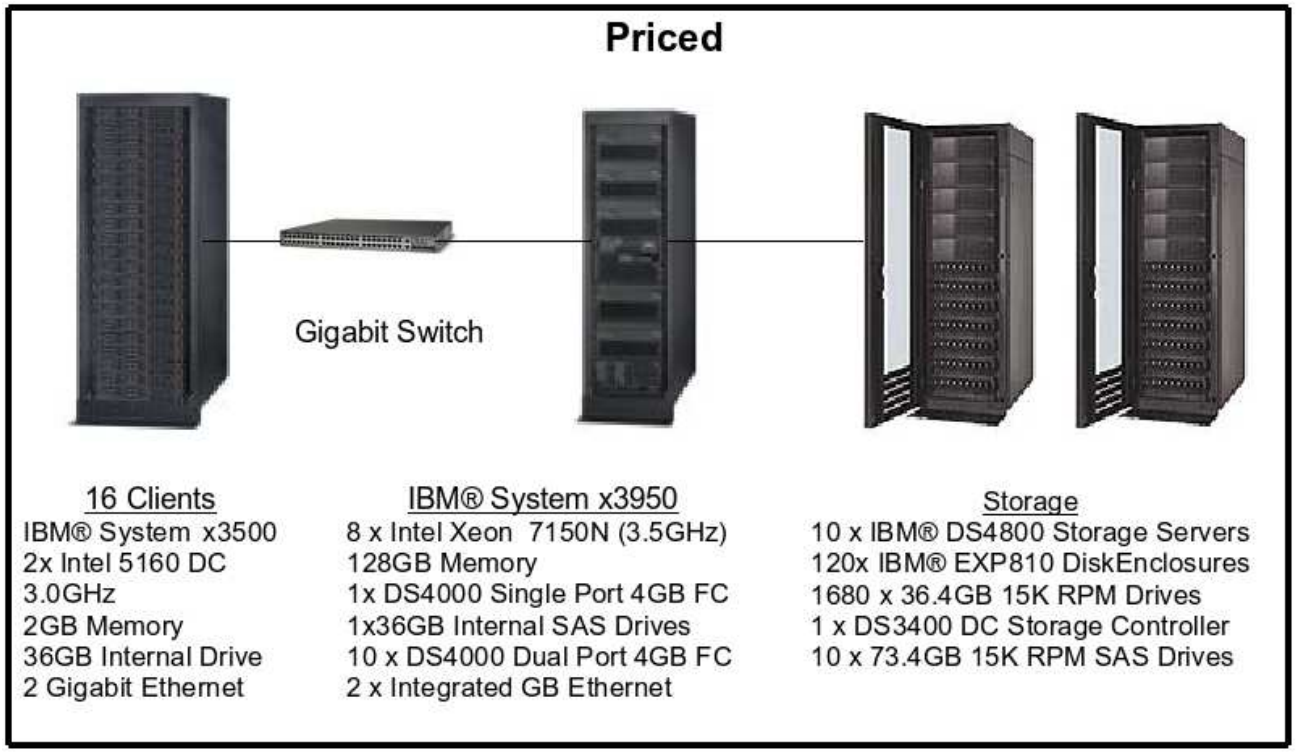
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test*
- *Number and type of disk units (and controllers, if applicable)*
- *Number of channels or bus connections to disk units, including the protocol type*
- *Number of LAN (e.g. Ethernet) connections, including routers, work stations, terminals, etc, that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8)*
- *Type and run-time execution location of software components (e.g. DBMS, client processes, transaction monitors, software drivers, etc)*

IBM System x3950 Benchmark Configuration



IBM System x3950 Priced Configuration



The measured configuration consisted of 1 x3500 client with two 3.0GHz Intel 5160 Dual Core processors and 15 x335 clients each with two 3.2GHz Intel Xeon DP processors. However, 16 x3500 clients were priced. Based on the response times and throughput values collected from each of the clients during the run and submitted to the auditor, the auditor determined that this upgrade does not have a material effect on the reported performance. All other components, including the Fibre Channel disk subsystem, adapters, HBAs, and the system under test, were priced as measured. For the priced configuration, see the Executive Summary.

1 Clause 1: Logical Data Base Design Related Items

1.1. Table Definitions

Listings must be provided for all table definition statements and all other statements used to setup the data base.

Appendix C contains the table definitions and the database load programs used to build the data base.

1.2. Database Organization

The physical organization of tables and indices, within the data base, must be disclosed.

Physical space was allocated to DB2 on the server disks according to the details provided in Appendix C.

1.3. Insert and/or Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT data base implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows.

There were no restrictions on insert and/or delete operations to any of the tables. The space required for an additional five percent of the initial table cardinality was allocated to DB2 and priced as static space.

The insert and delete functions were verified by the auditor. In addition, the auditor verified that the primary key for each database table could be updated outside the range of its initial partition.

1.4. Horizontal or Vertical Partitioning

While there are few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark, any such partitioning must be disclosed.

WAREHOUSE, DISTRICT, STOCK, CUSTOMER, HISTORY, ORDERS, ORDERLINE, and NEWORDER were horizontally partitioned into multiple tables.

Each table partition contains data associated with a range of 520 warehouses.

For each partitioned table, a view was created over all table partitions to provide full transparency of data manipulation.

No tables were replicated.

2 Clause 2: Transaction & Terminal Profiles Related Items

2.1. Verification for the Random Number Generator

The method of verification for the random number generation must be disclosed.

The seeds for each user were captured and verified by the auditor to be unique. In addition, the contents of the database were systematically searched and randomly sampled by the auditor for patterns that would indicate the random number generator had affected any kind of a discernible pattern; none were found.

2.2. Input/Output Screens

The actual layouts of the terminal input/output screens must be disclosed.

The screen layouts are now presented in HTML 1.0 web pages. Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C specifications were used as guidelines for html character placement.

2.3. Priced Terminal Features

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The emulated workstations, IBM eServer x3500 systems, are commercially available and support all of the requirements in Clause 2.2.2.4.

2.4. Presentation Managers

Any usage of presentation managers or intelligent terminals must be explained.

The workstations did not involve screen presentations, message bundling or local storage of TPC-C rows. All screen processing was handled by the client system. All data manipulation was handled by the server system.

2.5. Home and Remote Order-lines

The percentage of home and remote order-lines in the New-Order transactions must be disclosed.

Table 2-1 shows the percentage of home and remote transactions that occurred during the measurement period for the New-Order transactions.

2.6. New-Order Rollback Transactions

The percentage of New-Order transactions that were rolled back as a result of an illegal item number must be disclosed.

Table 2-1 shows the percentage of New-Order transactions that were rolled back due to an illegal item being entered.

2.7. Number of Items per Order

The number of items per order entered by New-Order transactions must be disclosed.

Table 2-1 show the average number of items ordered per New-Order transaction.

2.8. Home and Remote Payment Transactions

The percentage of home and remote Payment transactions must be disclosed.

Table 2-1 shows the percentage of home and remote transactions that occurred during the measurement period for the Payment transactions.

2.9. Non-Primary Key Transactions

The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the data base must be disclosed.

Table 2-1 shows the percentage of non-primary key accesses to the data base by the Payment and Order-Status transactions.

2.10. Skipped Delivery Transactions

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.

Table 2-1 shows the percentage of Delivery transactions missed due to a shortage of supply of rows in the NEW-ORDER table.

2.11. Mix of Transaction Types

The mix (i.e. percentages) of transaction types seen by the SUT must be disclosed.

Table 2-1 shows the mix percentage for each of the transaction types executed by the SUT.

New Order	**IBM System x3950**
Percentage of Home order lines	99.00%
Percentage of Remote order lines	1.00%
Percentage of Rolled Back Transactions	1.00%
Average Number of Items per order	10
Payment	
Percentage of Home transactions	85.00%
Percentage of Remote transactions	15.00%
Non-Primary Key Access	
Percentage of Payment using C_LAST	60.00%
Percentage of Order-Status using C_LAST	59.99%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.96%
Payment	43.02%
Order-Status	4.01%
Delivery	4.01%
Stock-Level	4.01%

Table 2-1: Numerical Quantities for Transaction and Terminal Profiles

2.12. Queuing Mechanism of Delivery

The queuing mechanism used to defer execution of the Delivery transaction must be disclosed.

The Delivery transaction was submitted to an ISAPI queue that is separate from the COM+ queue that the other transactions used. This queue is serviced by a variable amount of threads that are separate from the worker threads inside the web server. Web server threads are able to complete the on-line part of the Delivery transaction and immediately return successful queuing responses to the drivers. The threads servicing the queue are responsible for completing the deferred part of the transaction asynchronously.

3 Clause 3: Transaction and System Properties

The results of the ACID test must be disclosed along with a description of how the ACID requirements were met.

All ACID tests were conducted according to specification.

3.1. Atomicity Requirements

The system under test must guarantee that data base transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially-completed operations leave any effects on the data.

3.1.1. Atomicity of Completed Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately.

The following steps were performed to verify the Atomicity of completed transactions.

1. The balance, BALANCE_1, was retrieved from the CUSTOMER table for a random Customer, District and Warehouse combination.
2. The Payment transaction was executed and committed for the Customer, District, and Warehouse combination used in step 1.
3. The balance, BALANCE_2, was retrieved again for the Customer, District, and Warehouse combination used in step 1 and step 2. It was verified that BALANCE_1 was greater than BALANCE_2 by the amount of the Payment transaction.

3.1.2. Atomicity of Aborted Transactions

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have NOT been changed.

The following steps were performed to verify the Atomicity of the aborted Payment transaction:

1. The Payment application code was implemented with a Perl script that allowed the transaction to be rolled back rather than committed.
2. The balance, BALANCE_3, was retrieved from the Customer table for the same Customer, District, and Warehouse combination used in the completed Payment transaction Atomicity test.
3. The Payment transaction was executed for the Customer, District and Warehouse used in step 2. Rather than commit the transaction, the transaction was rolled back.
4. The balance, BALANCE_4 was retrieved again for the Customer, District, and Warehouse combination used in step 2. It was verified that BALANCE_4 was equal to BALANCE_3, demonstrating that there were no remaining effects of the rolled back Payment transaction.

3.2. Consistency Requirements

Consistency is the property of the application that requires any execution of a data base transaction to take the data base from one consistent state to another, assuming that the data base is initially in a consistent state.

Verify that the data base is initially consistent by verifying that it meets the consistency conditions defined in Clauses 3.3.2.1 to 3.3.2.4. Describe the steps used to do this in sufficient detail so that the steps are independently repeatable.

The specification defines 12 consistency conditions of which the following four are required to be explicitly demonstrated:

1. The sum of balances (d_ytd) for all Districts within a specific Warehouse is equal to the balance (w_ytd) of that Warehouse.
2. For each District within a Warehouse, the next available Order ID (d_next_o_id) minus one is equal to the most recent Order ID [max(o_id)] for the Order table associated with the preceeding District and Warehouse.

Additionally, that same relationship exists for the most recent Order ID [max(o_id)] for the New Order table associated with the same District and Warehouse. Those relationships can be illustrated as follows:

d_next_o_id – 1 = max(o_id) = max(no_o_id)

where (d_w_id = o_w_id = no_w_id) and (d_id = o_d_id = no_d_id)

3. For each District within a Warehouse, the value of the most recent Order ID [max(no_o_id)] minus the first Order ID [min(no_o_id)] plus one, for the New Order table associated with the District and Warehouse equals the number of rows in that New Order table. That relationship can be illustrated as follows:

   max(no_o_id) – min(no_o_id) + 1 = number of rows in New Order for the Warehouse/District

4. For each District within a Warehouse, the sum of Order Line counts [sum(o_ol_cnt)] for the Order table associated with the District equals the number of rows in the Order Line table associated with the same District. That relationship can be illustrated as follows:

   sum(o_ol_cnt) = number of rows in the Order Line table for the Warehouse/District

An RTE driven run was executed against a freshly loaded database. After the run the 4 consistency conditions defined above were tested using a script to issue queries to the database. All queries showed that the database was still in a consistent state.

3.3. Isolation Requirements

Operations of concurrent data base transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

The benchmark specification defines nine tests to demonstrate the property of transaction isolation. The tests, described in Clauses 3.4.2.1 – 3.4.2.9 were all successfully executed using a series of scripts. Case A was observed during the execution of Isolation Tests 7-9.

3.4. Durability Requirements

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure data base consistency after recovery from any one of the failures listed in Clause 3.5.3

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data (this test includes failure of all or part of memory)*
- *Instantaneous interruption (system crash/system hang) in processing that requires system reboot to recover*
- *Failure of all or part of memory (loss of contents)*

Failure of Log Disk, Log Fast Write Cache, Instantaneous Interruption and Memory Failure:

This test was conducted on a fully-scaled database. The following steps were successfully performed:

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full load test was started and allowed to run for over 10 minutes.
3. One of the disks containing the transaction log was removed. Since the log was implemented as a RAID-5 array, DB2 continued to process the transactions successfully.
4. The test continued for at least another 5 minutes.
5. A storage controller holding one copy of the mirrored write cache for the log was removed from the storage subsystem. The contents of the write cache mirrors became out-of-sync.

6. The system was subsequently powered off, which removed power from all system components, including memory.
7. The storage controller from step 5 was reinserted into the storage subsystem. The controller detected the cache out-of-sync condition and synchronized with the write cache mirror in the other controller.
8. The disk from step 3 was replaced.
9. The system was powered back on and DB2 was allowed to recover.
10. Step 1 was performed returning the value for SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.
11. Consistency condition 3 was verified.

Failure of Durable Medium Containing TPC-C Database Tables:

This test was conducted on a fully-scaled database. The following steps were successfully performed:

1. The contents of the database were backed up in full.
2. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
3. A scaled-down test was started with 12.5% of the full load.
4. A disk containing the TPC-C tables was removed, causing DB2 to report numerous errors.
5. The system was subsequently shutdown.
6. The disk was reinserted.
7. The system was powered back on.
8. The full database was restored from the backup copy in step 1.
9. DB2 was restarted and the transactions in the log were applied to the database.
10. Step 2 was performed returning SUM_2. It was verified that SUM_2 was equal to SUM_1 plus the number of completed New_Order transactions recorded by the RTE.
11. Consistency condition 3 was verified.

4 Clause 4: Scaling and Data Base Population Related Items

4.1. Cardinality of Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed.

Table 4-1 portrays the TPC Benchmark™ C defined tables and the number of rows for each table as they were built initially.

Table Name	Number of Rows
Warehouse	41,600
District	416,000
Customer	1,248,000,000
History	1,248,000,000
Order	1,248,000,000
New Order	374,400,000
Order Line	12,479,999,000
Stock	4,160,000,000
Item	100,000

Table 4-1: Initial Cardinality of Tables

4.2. Distribution of Tables and Logs

The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems.

There is one Logical Disk (LD) for the logs.

There is one storage adapter for the log LD.

The log LD is configured as a RAID5 (9+p) disk array, with 10 physical disks. Each physical disk has a capacity of 73.4 GB. The total Raid 5 capacity available is 660.6 GB.

There are 80 Logical Disks (LD) for the tables.

There are 10 storage adapters for the tables. Each adapter is assigned 8 LDs.

Each LD is configured as a RAID0 disk array, with 21 physical disks.

There is a total of 1680 data disks and each physical disk has a capacity of 36.4 GB drives.

Each LD is partitioned identically. Each LD contains 14 partitions, 12 of which are used for DB2 Containers. Partitions are laid out on an LD as follows:

Partition#	Blocks	Container Usage
1	7968	Warehouse
2	8001	District

3 7969 Item
4 Extended partition
5 18450558 Stock
6 13253559 Customer
7 963827 Customer Index
8 1445773 History
9 963849 Orders
10 803195 Order Index
11 21687681 Order Line
12 321237 New OrdersA
13 321237 New OrdersB
14 677265552 Not used for data tables

Tablespaces are laid out to use LDs as follows:

Warehouse 80 tablespaces each using 1 LD
District 80 tablespaces each using 1 LD
Item One tablespace using 80 LDs
Stock 80 tablespaces each using 1 LD
Customer 80 tablespaces each using 1 LD
Customer Index 80 tablespaces each using 1 LD
History 80 tablespaces each using 1 LD
Orders 80 tablespaces each using 1 LD
Order Index 80 tablespaces each using 1 LD
Order Line 80 tablespaces each using 1 LD
New OrdersA 80 tablespaces each using 1 LD
New OrdersB 80 tablespaces each using 1 LD

4.3. Data Base Model Implemented

A statement must be provided that describes the data base model implemented by the DBMS used.

The database manager used for this testing was DB2 9. DB2 is a relational DBMS. DB2 remote stored procedures and embedded SQL statements were used. The DB2 stored procedures were invoked via SQL CALL statements. Both the client application and stored procedures were written in embedded C code.

4.4. Partitions/Replications Mapping

The mapping of data base partitions/replications must be explicitly described.

The specifics of the distribution of partitioned and non-partitioned tables across the physical media can be found in tables 4-2.

Data Distribution Logical Disks (LDs)			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD8 Partition 1	FC1	raw1 –raw8	ts_wh_01-ts_wh_08
LD9 – LD16 Partition 1	FC2	raw9 –raw16	ts_wh_09-ts_wh_16
LD17 – LD24 Partition 1	FC3	raw17 –raw24	ts_wh_17-ts_wh_24
LD25 – LD32 Partition 1	FC4	raw25 –raw32	ts_wh_25-ts_wh_32
LD33 – LD40 Partition 1	FC5	raw33 –raw40	ts_wh_33_ts_wh_40
LD41 – LD48 Partition 1	FC6	raw41 –raw48	ts_wh_41_ts_wh_48

LD49 – LD56 Partition 1	FC7	raw49 –raw56	ts_wh_49_ts_wh_56
LD57 – LD64 Partition 1	FC8	raw57 –raw64	ts_wh_57_ts_wh_64
LD65 – LD72 Partition 1	FC9	raw65 –raw72	ts_wh_65_ts_wh_72
LD73 – LD80 Partition 1	FC10	raw73 –raw80	ts_wh_73_ts_wh_80
LD1 – LD8 Partition 2	FC1	raw101 –raw108	ts_dis_01-ts_dis_08
LD9 – LD16 Partition 2	FC2	raw109 –raw116	ts_dis_09-ts_dis_16
LD17 – LD24 Partition 2	FC3	raw117 –raw124	ts_dis_17-ts_dis_24
LD25 – LD32 Partition 2	FC4	raw125 –raw132	ts_dis_25-ts_dis_32
LD33 – LD40 Partition 2	FC5	raw133–raw140	ts_dis_33-ts_dis_40
LD41 – LD48 Partition 2	FC6	raw141 –raw148	ts_dis_41_ts_dis_48
LD49 – LD56 Partition 2	FC7	raw149 –raw156	ts_dis_49_ts_dis_56
LD57 – LD64 Partition 2	FC8	raw157 –raw164	ts_dis_57_ts_dis_64
LD65 – LD72 Partition 2	FC9	raw165 –raw172	ts_dis_65_ts_dis_72
LD73 – LD80 Partition 2	FC10	raw173 –raw180	ts_dis_73_ts_dis_80
LD1 – LD8 Partition 3	FC1	raw201 –raw208	ts_item
LD9 – LD16 Partition 3	FC2	raw209 –raw216	ts_item
LD17 – LD24 Partition 3	FC3	raw217 –raw224	ts_item
LD25 – LD32 Partition 3	FC4	raw225 –raw232	ts_item
LD33 – LD40 Partition 3	FC5	raw233–raw240	ts_item
LD41 – LD48 Partition 3	FC6	raw241 –raw248	ts_item
LD49 – LD56 Partition 3	FC7	raw249 –raw256	ts_item
LD57 – LD64 Partition 3	FC8	raw257 –raw264	ts_item
LD65 – LD72 Partition 3	FC9	raw265 –raw272	ts_item
LD73 – LD80 Partition 3	FC10	raw273 –raw280	ts_item
LD1 – LD8 Partition 5	FC1	raw301 –raw308	ts_stock_01-ts_stock_08
LD9 – LD16 Partition 5	FC2	raw309 –raw316	ts_stock_09-ts_stock_16
LD17 – LD24 Partition 5	FC3	raw317 –raw324	ts_stock_17-ts_stock_24
LD25 – LD32 Partition 5	FC4	raw325 –raw332	ts_stock_25-ts_stock_32
LD33 – LD40 Partition 5	FC5	raw333–raw340	ts_stock_33-ts_stock_40
LD41 – LD48 Partition 5	FC6	raw341 –raw348	ts_stock_41-ts_stock_48
LD49 – LD56 Partition 5	FC7	raw349 –raw356	ts_stock_49_ts_stock_56
LD57 – LD64 Partition 5	FC8	raw357 –raw364	ts_stock_57_ts_stock_64
LD65 – LD72 Partition 5	FC9	raw365 –raw372	ts_stock_65_ts_stock_72
LD73 – LD80 Partition 5	FC10	raw373 –raw380	ts_stock_73_ts_stock_80

LD1 – LD8 Partition 6	FC1	raw401 –raw408	ts_customer_01-ts_customer_08
LD9 – LD16 Partition 6	FC2	raw409 –raw416	ts_customer_09-ts_customer_16
LD17 – LD24 Partition 6	FC3	raw417 –raw424	ts_customer_17-ts_customer_24
LD25 – LD32 Partition 6	FC4	raw425 –raw432	ts_customer_25-ts_customer_32
LD33 – LD40 Partition 6	FC5	raw433–raw440	ts_customer_33-ts_customer_40
LD41 – LD48 Partition 6	FC6	raw441 –raw448	ts_customer_41-ts_customer_48
LD49 – LD56 Partition 6	FC7	raw449 –raw456	ts_customer_49_ts_customer_56
LD57 – LD64 Partition 6	FC8	raw457 –raw464	ts_customer_57_ts_customer_64
LD65 – LD72 Partition 6	FC9	raw465 –raw472	ts_customer_65_ts_customer_72
LD73 – LD80 Partition 6	FC10	raw473 –raw480	ts_customer_73_ts_customer_80
LD1 – LD8 Partition 7	FC1	raw501 –raw508	is_customer_01-is_customer_08
LD9 – LD16 Partition 7	FC2	raw509 –raw516	is_customer_09-is_customer_16
LD17 – LD24 Partition 7	FC3	raw517 –raw524	is_customer_17-is_customer_24
LD25 – LD32 Partition 7	FC4	raw525 –raw532	is_customer_25-is_customer_32
LD33 – LD40 Partition 7	FC5	raw533–raw540	is_customer_33-is_customer_40
LD41 – LD48 Partition 7	FC6	raw541 –raw548	is_customer_41-is_customer_48
LD49 – LD56 Partition 7	FC7	raw549 –raw556	is_customer_49_is_customer_56
LD57 – LD64 Partition 7	FC8	raw557 –raw564	is_customer_57_is_customer_64
LD65 – LD72 Partition 7	FC9	raw565 –raw572	is_customer_65_is_customer_72
LD73 – LD80 Partition 7	FC10	raw573 –raw580	is_customer_73_is_customer_80
LD1 – LD8 Partition 8	FC1	raw601 –raw608	ts_history_01-ts_history_08
LD9 – LD16 Partition 8	FC2	raw609 –raw616	ts_history_09-ts_history_16
LD17 – LD24 Partition 8	FC3	raw617 –raw624	ts_history_17-ts_history_24
LD25 – LD32 Partition 8	FC4	raw625 –raw632	ts_history_25-ts_history_32
LD33 – LD40 Partition 8	FC5	raw633–raw640	ts_history_33-ts_history_40
LD41 – LD48 Partition 8	FC6	raw641 –raw648	ts_history_41-ts_history_48
LD49 – LD56 Partition 8	FC7	raw649 –raw656	ts_history_49_ts_history_56
LD57 – LD64 Partition 8	FC8	raw657 –raw664	ts_history_57_ts_history_64
LD65 – LD72 Partition 8	FC9	raw665 –raw672	ts_history_65_ts_history_72
LD73 – LD80 Partition 8	FC10	raw673 –raw680	ts_history_73_ts_history_80
LD1 – LD8 Partition 9	FC1	raw701 –raw708	ts_order_01-ts_order_08
LD9 – LD16 Partition 9	FC2	raw709 –raw716	ts_order_09-ts_order_16
LD17 – LD24 Partition 9	FC3	raw717 –raw724	ts_order_17-ts_order_24
LD25 – LD32 Partition 9	FC4	raw725 –raw732	ts_order_25-ts_order_32

LD33 – LD40 Partition 9	FC5	raw733–raw740	ts_order_33-ts_order_40
LD41 – LD48 Partition 9	FC6	raw741 –raw748	ts_order_41-ts_order_48
LD49 – LD56 Partition 9	FC7	raw749 –raw756	ts_order_49_ts_order_56
LD57 – LD64 Partition 9	FC8	raw757 –raw764	ts_order_57_ts_order_64
LD65 – LD72 Partition 9	FC9	raw765 –raw772	ts_order_65_ts_order_72
LD73 – LD80 Partition 9	FC10	raw773 –raw780	ts_order_73_ts_order_80
LD1 – LD8 Partition 10	FC1	raw801 –raw808	is_order_01-is_order_08
LD9 – LD16 Partition 10	FC2	raw809 –raw816	is_order_09-is_order_16
LD17 – LD24 Partition 10	FC3	raw817 –raw824	is_order_17-is_order_24
LD25 – LD32 Partition 10	FC4	raw825 –raw832	is_order_25-is_order_32
LD33 – LD40 Partition 10	FC5	raw833–raw840	is_order_33-is_order_40
LD41 – LD48 Partition 10	FC6	raw841 –raw848	is_order_41-is_order_48
LD49 – LD56 Partition 10	FC7	raw849 –raw856	is_order_49_is_order_56
LD57 – LD64 Partition 10	FC8	raw857 –raw864	is_order_57_is_order_64
LD65 – LD72 Partition 10	FC9	raw865 –raw872	is_order_65_is_order_72
LD73 – LD80 Partition 10	FC10	raw873 –raw880	is_order_73_is_order_80
LD1 – LD8 Partition 11	FC1	raw901 –raw908	ts_orderline_01-ts_orderline_08
LD9 – LD16 Partition 11	FC2	raw909 –raw916	ts_orderline_09-ts_orderline_16
LD17 – LD24 Partition 11	FC3	raw917 –raw924	ts_orderline_17-ts_orderline_24
LD25 – LD32 Partition 11	FC4	raw925 –raw932	ts_orderline_25-ts_orderline_32
LD33 – LD40 Partition 11	FC5	raw933–raw940	ts_orderline_33-ts_orderline_40
LD41 – LD48 Partition 11	FC6	raw941 –raw948	ts_orderline_41-ts_orderline_48
LD49 – LD56 Partition 11	FC7	raw949 –raw956	ts_orderline_49_ts_orderline_56
LD57 – LD64 Partition 11	FC8	raw957 –raw964	ts_orderline_57_ts_orderline_64
LD65 – LD72 Partition 11	FC9	raw965 –raw972	ts_orderline_65_ts_orderline_72
LD73 – LD80 Partition 11	FC10	raw973 –raw980	ts_orderline_73_ts_orderline_80
LD1 – LD8 Partition 12	FC1	raw1001 – raw1008	ts_newordA_01-ts_newordA_08
LD9 – LD16 Partition 12	FC2	raw1009 – raw1016	ts_newordA_09-ts_newordA_16
LD17 – LD24 Partition 12	FC3	raw1017 – raw1024	ts_newordA_17-ts_newordA_24
LD25 – LD32 Partition 12	FC4	raw1025 – raw1032	ts_newordA_25-ts_newordA_32
LD33 – LD40 Partition 12	FC5	raw1033–raw1040	ts_newordA_33-ts_newordA_40
LD41 – LD48 Partition 12	FC6	raw1041 – raw1048	ts_newordA_41-ts_newordA_48

LD49 – LD56 Partition 12	FC7	raw1049 – raw1056	ts_newordA_49_ts_newordA_56
LD57 – LD64 Partition 12	FC8	raw1057 – raw1064	ts_newordA_57_ts_newordA_64
LD65 – LD72 Partition 12	FC9	raw1065 – raw1072	ts_newordA_65_ts_newordA_72
LD73 – LD80 Partition 12	FC10	raw1073 – raw1080	ts_newordA_73_ts_newordA_80
LD1 – LD8 Partition 13	FC1	raw1141 – raw1148	ts_newordB_01-ts_newordB_08
LD9 – LD16 Partition 13	FC2	raw1149 – raw1156	ts_newordB_09-ts_newordB_16
LD17 – LD24 Partition 13	FC3	raw1157 – raw1164	ts_newordB_17-ts_newordB_24
LD25 – LD32 Partition 13	FC4	raw1165 – raw1172	ts_newordB_25-ts_newordB_32
LD33 – LD40 Partition 13	FC5	raw1173 – raw1180	ts_newordB_33-ts_newordB_40
LD41 – LD48 Partition 13	FC6	raw1141 – raw1148	ts_newordB_41-ts_newordB_48
LD49 – LD56 Partition 12	FC7	raw1149 – raw1156	ts_newordB_49_ts_newordB_56
LD57 – LD64 Partition 12	FC8	raw1157 – raw1164	ts_newordB_57_ts_newordB_64
LD65 – LD72 Partition 12	FC9	raw1165 – raw1172	ts_newordB_65_ts_newordB_72
LD73 – LD80 Partition 12	FC10	raw1173 – raw1180	ts_newordB_73_ts_newordB_80

Table 4-2: IBM System x3950 Data Distribution Benchmark Configuration

4.5. 60-Day Space Calculations

Details of the 60 day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed

All data sizes in MB unless otherwise stated					
Warehouses	41,600				
Measured TpmC	511,342				
Table	**Rows**	**Table**	**Index**	**5% Space**	**Total Space**
Warehouse	41,600	50	0	3	53
District	416,000	90	0	5	95
Item	100,000	10	0	1	11
Stock	4,160,000,000	1,354,250	0	67,713	1,421,963
Customer	1,248,000,000	975,080	60,100	51,759	1,086,939
New-Order	374,400,000	28,800	0	1,440	30,240
Orders	1,248,000,000	47,844	34,940	0	82,784
Order-Line	12,480,000,000	836,140	0	0	836,140
History	1,248,000,000	76,960	0	0	76,960
Additional Overhead		719,496			719,496
Free Space	136,691				
Dynamic Space	960,944		30 Minute log Computations		
Static Space	3,293,735		Log Written (KB)		36,356,416
Daily Growth	188,989		New-Order Txns		15,340,260
Daily Spread	0		Log Written per New-Order (KB)		2.37
Data Storage Requirement					
60 Days (MB)	14,633,070				
60 Days (GB)	14,290				
Log Storage Requirement					
8 Hours (GB)	554.75				
Disk Sizing					
	Formatted	**SUT**		**Priced**	
Disk Type	**Capacity (GB)**	**# of Disks**	**Capacity (GB)**	**# of Disks**	**Capacity (GB)**
DB FastT 36.4GB	36.40	1,680	61,152	1,680	61,152
LOG FastT RAID5	73.40	9	661	9	661
OS SCSI 36GB	36.40	1	36	1	36
Total Capacity					61,849

5 Clause 5: Performance Metrics and Response Time Related Items

5.1. Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the Menu response time.

Table 5-1 lists the response times and the ninetieth percentiles for each of the transaction types for the measured system.

5.2. Keying and Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type.

Table 5-1 lists the TPC-C keying and think times for the measured system.

Response Times	New Order	Payment	Order Status	Delivery (int./def.)	Stock Level	Menus
90 %	1.34	1.36	1.14	0.48/0.38	1.76	0.63
Average	0.70	0.68	0.65	0.21/0.22	1.16	0.23
Maximum	11.29	11.16	10.97	9.89/1.19	12.0	9.89
Think Times						
Minimum	0	0	0	0	0	N/A
Average	12.04	12.04	10.04	5.04	5.04	N/A
Maximum	120.31	120.33	100.31	50.31	50.31	N/A
Keying Times						
Minimum	18.00	3.00	2.00	2.00	2.00	N/A
Average	18.00	3.00	2.00	2.00	2.00	N/A
Maximum	18.02	3.02	2.02	2.02	2.02	N/A

Table 5-1: Think and Keying Times

5.3. Response Time Frequency Distribution

Response time frequency distribution curves must be reported for each transaction type.

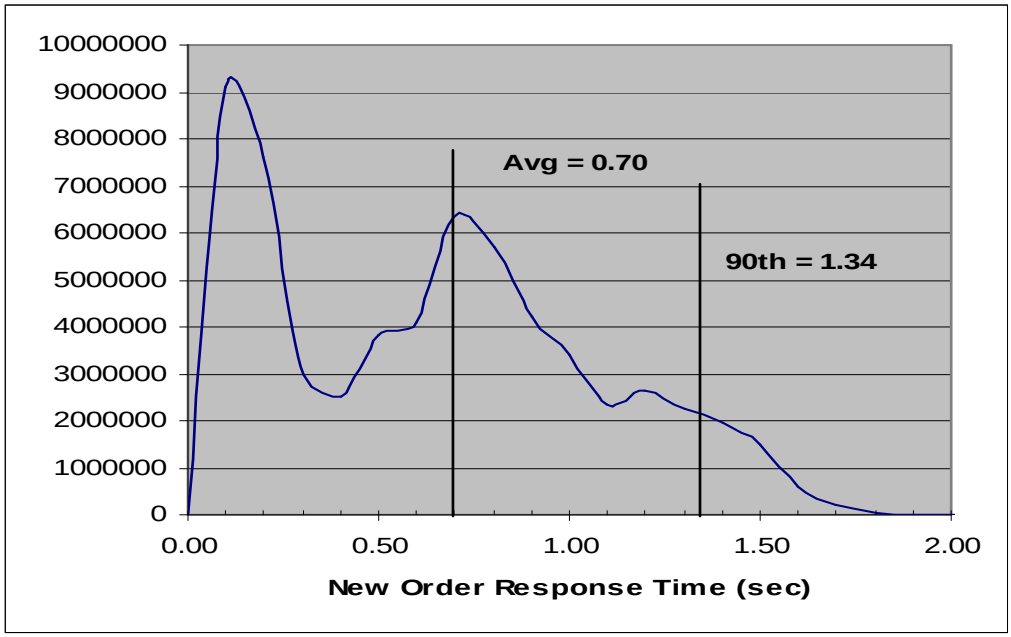


Figure 5-1: New-Order Response Time Distribution

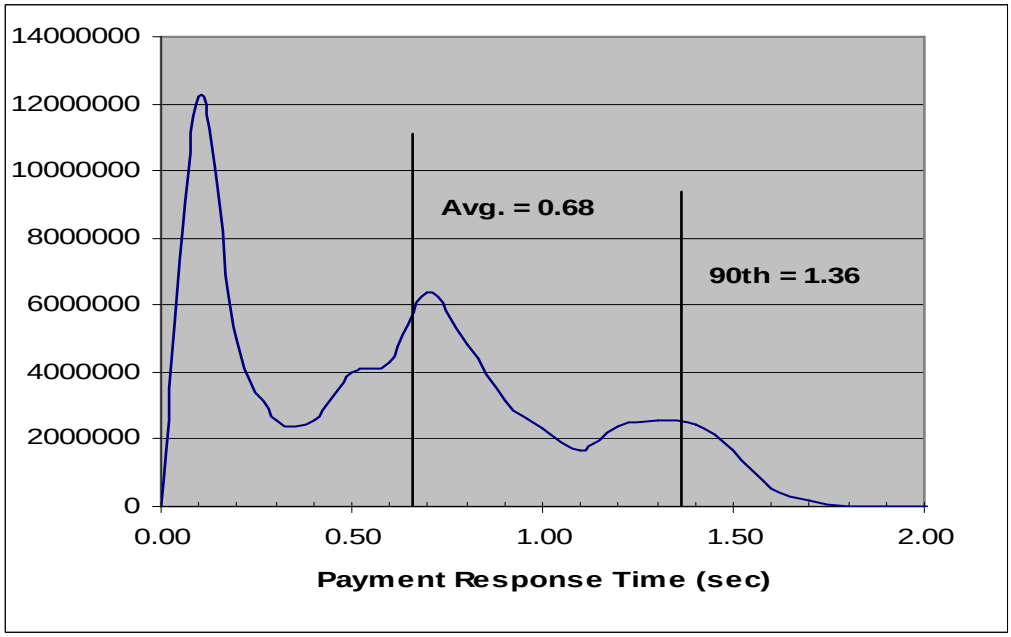


Figure 5-2: Payment Response Time Distribution

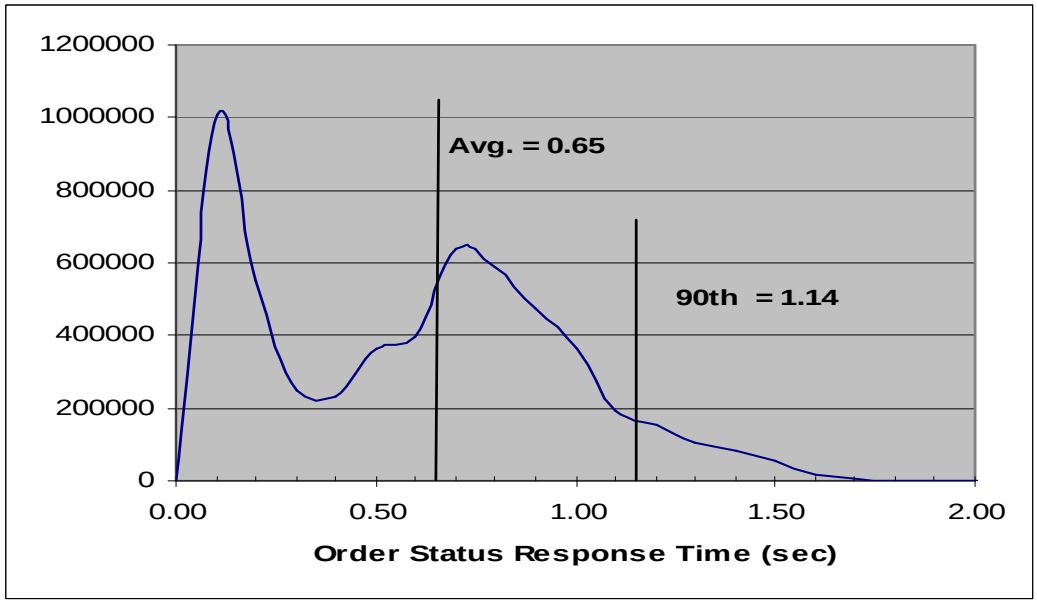


Figure 5-3: Order-Status Response Time Distribution

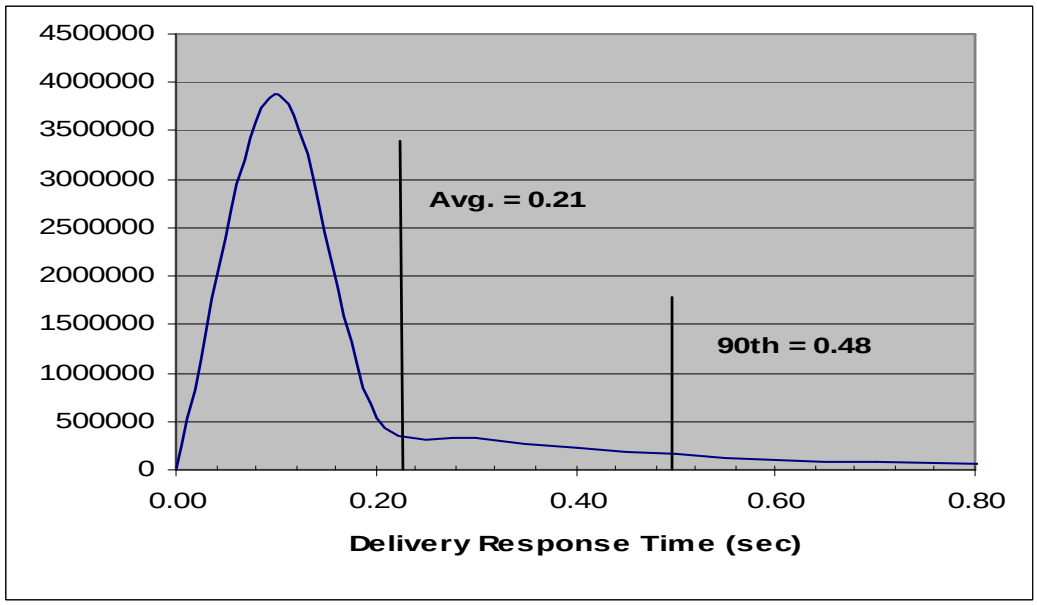


Figure 5-4: Delivery (Interactive) Response Time Distribution

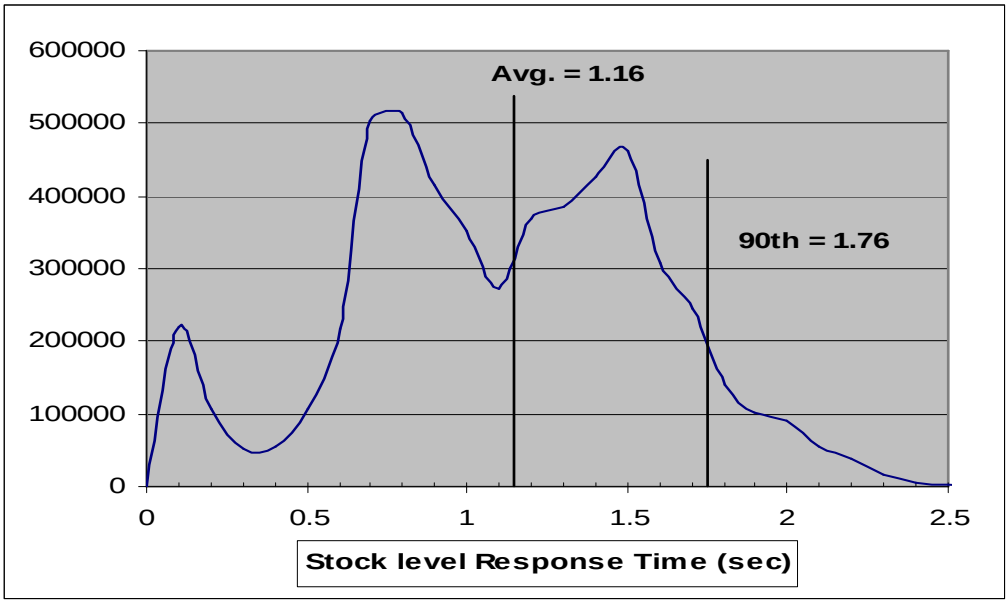


Figure 5-5: Stock Level Response Time Distribution

5.4. Performance Curve for Response Time versus Throughput

The performance curve for response times versus throughput must be reported for the New-Order transaction.

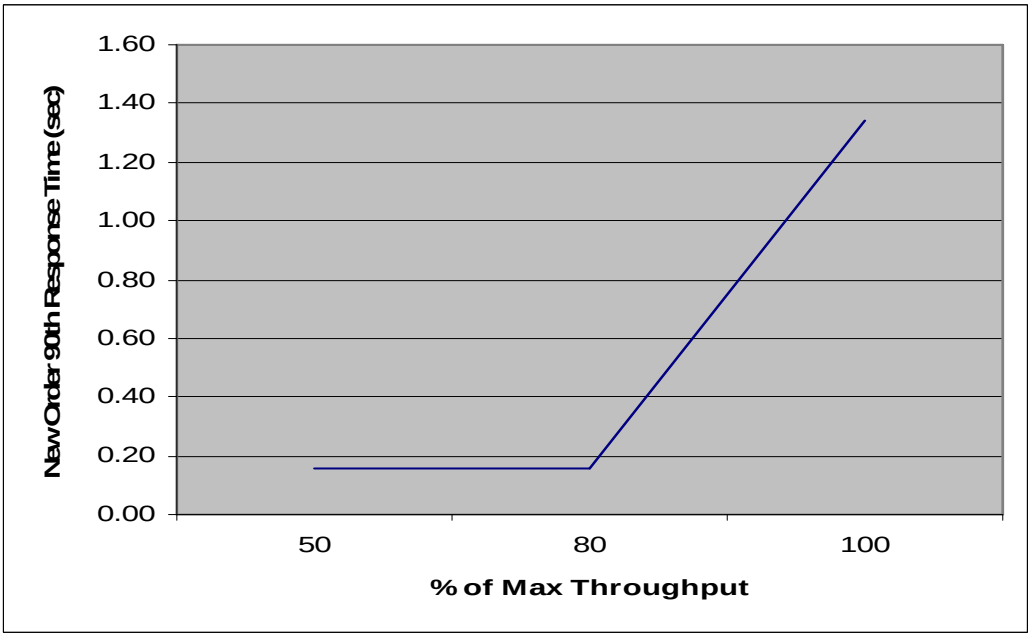


Figure 5-6: New-Order Response Time vs. Throughput

5.5. Think Time Frequency Distribution

A graph of the think time frequency distribution must be reported for the New-Order transaction.

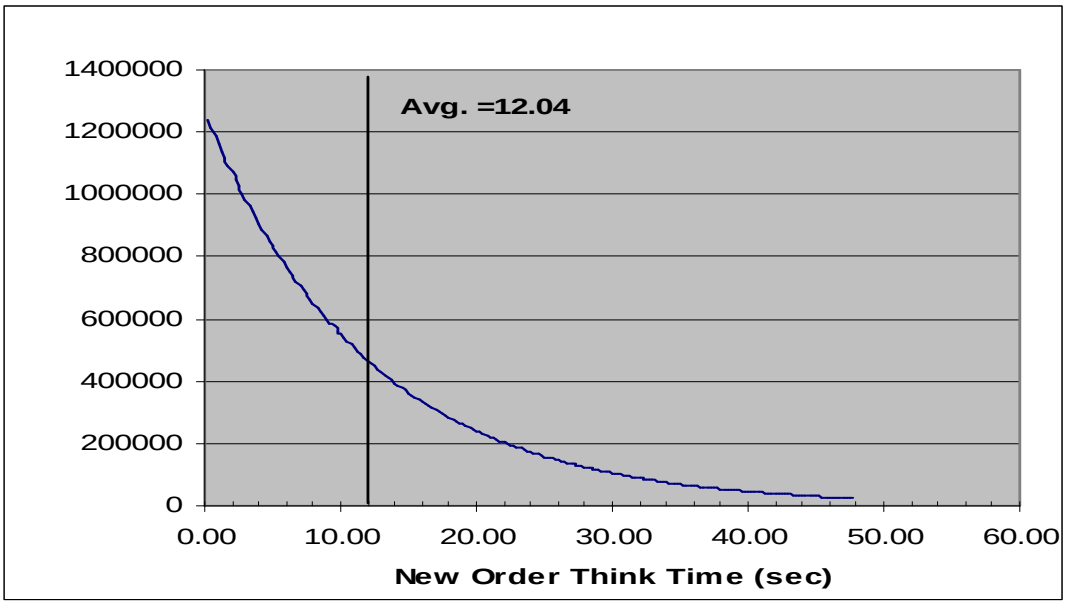


Figure 5-7: New-Order Think Time Distribution

5.6. Throughput versus Elapsed Time

A graph of throughput versus elapsed time must be reported for the New-Order transaction.

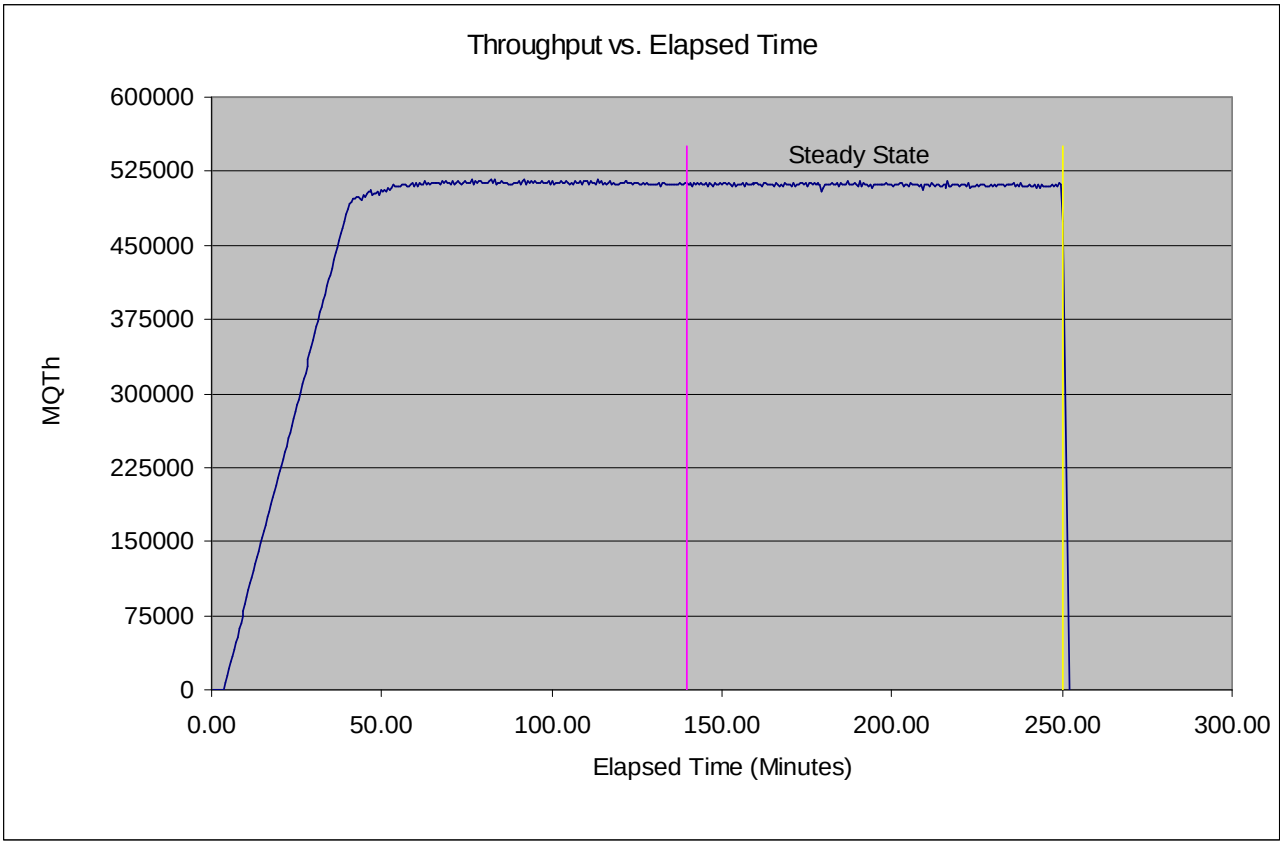


Figure 5-8: New-Order Throughput vs. Elapsed Time

5.7. Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be described.

All the emulated users were allowed to logon and do transactions. The user ramp-up phase is clearly visible on the graph above. Refer to the Numerical Quantities Summary pages for the rampup time. Figure 5-8 New-Order throughput versus Elapsed Time graph shows that the system maintained a steady state during the measurement interval

5.8. Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example check pointing, writing redo/undo log records, etc), actually occurred during the measurement interval must be reported.

A 2-hour measurement interval was used to guarantee that all work normally performed during an 8-hour sustained test is included in the reported throughput.

5.8.1. Transaction Flow

Each of the 4 (non-delivery) transactions is serviced by 2 individual programs, Internet Information System 6.0 (IIS) and a Microsoft COM+ 1.0 Queued Component Server, used as the transaction manager (COM+). Both programs are running on the client system:

- The initial HTML 1.0 request is serviced by an ISAPI custom-written handler running on Internet Information System 6.0. IIS is responsible for handling all HTML requests. The web server communicates to the COM+ server through a Microsoft COM+ API interface.
- COM+ communicates with the server system over Ethernet and handles all database operations**,** using DB2 embedded SQL calls.
  When the COM+ server boots up, it creates a configurable amount of connections to the server (listed in application settings).
  COM+ routes the transaction and balances the load according to the options defined in the Component Services GUI for the COM+ server application and settings in the Windows 2003 registry. The configuration file and registry variables are listed in Appendix B.2.
  At the beginning, each TPC-C user sends a pair of HTML 1.0 requests submitting its unique warehouse and district to the IIS ISAPI handler. Upon successful validation of user's login, IIS the displays an HTML form which encapsulates the TPC-C transaction menu.
  The transaction flow is described below:
  - The TPC-C user requests the transaction type's HTML form and proceeds to generate (fill in) a GET request with the required files for the transaction.
  - IIS accepts the filled in GET request , parses, and validates all values entered by the user.
  - It then proceeds to transmit those values to the COM+ server through an transaction type specific COM+ api interface.
  - The COM+ Pool Manager receives the request and first decides if there is a connection object in the pool available to service it.
    If so, the connection is used to send the transaction request to the Server.
    If no connection is available, the request will enter a COM+ internal queue and will be serviced by the next available connection.
  - Once the connection is available to be used, a COM+ pool thread receives the transaction and calls a TPC-C back end DB2 client api to execute all database operations related to the transaction type. (All the transaction information entered on the HTML form is available in a data structure provided by the ISAPI caller).
  - The transaction is committed and the DB2 back end client returns control back to the COM pool thread.
  - COM pool thread returns control to the ISAPI caller.
    (All transaction results are inside the data structure that the ISAPI caller provided to the COM+ api in the parameter list).
  - ISAPI caller returns control to the "screen application" by doing a PUT request.

5.8.2. Database Transaction

All database operations are performed by the TPC-C back-end programs. The process is described below:

Using embedded SQL calls, the TPC-C back-end program interacts with DB2 Server to perform SQL data manipulations such as update, select, delete and insert, as required by the transaction. After all database operations are performed for a transaction, the transaction is committed.

DB2 Server proceeds to update the database as follows:

When DB2 Server changes a database table with an update, insert, or delete operation, the change is initially made in memory, not on disk. When there is not enough space in the memory buffer to read in or write additional data pages, DB2 Server will make space by flushing some modified pages to disk. Modified pages are also written to disk as part of the “Soft” checkpoint to ensure that no updates remain unflushed for longer than the allowed time. Before a change is made to the database, it is first recorded in the transaction log. This ensures that the database can be recovered completely in the event of a failure. Using the transaction log, transactions that started but did not complete prior to a failure can be undone, and transactions recorded as complete in the transaction log but not yet written to disk can be redone.

5.8.3. Checkpoints

DB2 uses a write-ahead-logging protocol to guarantee recovery. This protocol uses “Soft” checkpoint to write least-recently-used database pages to disk independent of transaction commit. However, enough log information to redo/undo the change to a database pages is committed to disk before the database page itself is written. This protocol therefore renders checkpoint unnecessary for DB2 . For a more detailed description of the general principles of the write-ahead-logging protocol, see the IBM research paper, “ARIES: A Transaction Recovery Method Supporting Fine Granularity

Locking and Partial Rollbacks Using Write-Ahead Logging," by C. Mohan, Database Technology Institute, IBM Almaden Research Center.
(http:// portal.acm.org/citation.cfm?id=128770&coll=portal&dl=ACM&CFID=10343790&CFTOKEN=42047146)

5.9. Measurement Interval

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included.

A 2-hour measurement interval was used. No connections were lost during the run.

6 Clause 6: SUT, Driver, and Communication Definition Related Items

6.1. RTE Availability

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs to the RTE had been used.

IBM used an internally developed RTE for these tests. 41,600 warehouses were configured. A rampup time of one hour twenry six minutes was specified, along with a run time of two hours.

6.2. Functionality and Performance of Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system.

No components were emulated.

6.3. Network Bandwidth

The bandwidth of the network(s) used in the tested/priced configuration must be disclosed.

The network between the clients and the database server was configured as 1000 MegaBits per second Full Duplex.

6.4. Operator Intervention

If the configuration requires operator intervention, the mechanism and the frequency of this intervention must be disclosed.

No operator intervention is required to sustain the reported throughput during the eight-hour period.

7 Clause 7: Pricing Related Items

7.1. Hardware and Programs Used

A detailed list of the hardware and software used in the priced system must be reported. Each item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, contents of the package must be disclosed. Pricing source(s) and effective date(s) must also be reported.

The detailed list of all hardware and programs for the priced configuration is listed in the pricing sheets for each system reported. The prices for all products and features that are provided by IBM are available the same day as product or feature availability.

7.2. Three Year Cost of System Configuration

The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The pricing details for this disclosure is contained in the executive summary pages. All 3rd party quotations are included at the end of this report in Appendix D. All prices are based on IBM US list prices.

Discount are based on US list prices and for similar quantities and configurations. A discount of 31.72% has been applied to specified IBM hardware, and services based on the total value and quantities of the components of the configuration.

7.3. Availability Dates

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

All components of the SUT will be available on March 26, 2007.

7.4. Statement of tpmC and Price/Performance

A statement of the measured tpmC, as well as the respective calculations for 3-year pricing, price/performance (price/tpmC), and the availability date must be disclosed.

System	tpmC	3-year System Cost	$/tpmC	Availability Date
IBM System x3950	511,342	$2,784,599 USD	$5.45 USD	March 26, 2007

Please refer to the price list on the Executive Summary page for details.

8 Clause 9: Audit Related Items

If the benchmark has been independently audited, then the auditor's name, address, phone number, and a brief audit summary report indicating compliance must be included in the Full Disclosure Report. A statement should be included, specifying when the complete audit report will become available and who to contact in order to obtain a copy.

The auditor's attestation letter is included in this section of this report:

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Raymond J. Venditti
IBM Linux Performance
11501 Burnet Road
Austin, TX 78758

Berni Schiefer
IBM DB2 Performance
8200 Warden Avenue
Markham, Ontario L6G1C7

March 16, 2007

I verified the TPC Benchmark™ C performance of the following Client Server configuration:

Platform: IBM System x3950 c/s
Operating system: SUSE Linux Enterprise Server 10 for Intel EM64T
Database Manager: IBM DB2 9
Transaction Manager: Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: IBM System x3950				
8 x Intel Xeon 7150N (3.5GHz)	128 GB (16 MB L3 cache/cpu)	1680 x 36.4 GB FC (ext.) 9 x 73.4 GB SAS (ext.) 1 x 36 GB SAS (int.)	1.34 Seconds	511,342.4
Sixteen Clients: IBM System x3500 (each with)				
2 x Intel Xeon 5160 DualCore (3.0 GHz)	2 GB main (2 MB L2 cache/cpu)	1 x 36 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

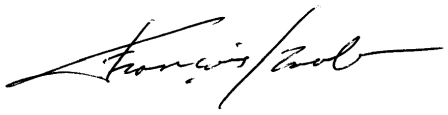
The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 120 minutes
- Write-ahead-logging was active during the measurement interval
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

> The measured configuration included fifteen IBM x335 client systems and one IBM x3500 client system. The IBM x335 systems were substituted, one for one, with IBM x3500 systems in the priced configuration. The TPC requirements for substitution of priced clients were verified and met.

Respectfully Yours,



François Raab, President

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

9 Appendix A: Client Server Code

9.1. Client/Terminal Handler Code

Makefile.config

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
######

#
# Makefile.config - NT/Win2000 Makefile Configuration
#

# Make Configuration (MSVC)
MAKE=nmake.exe

# Compiler Configuration (MSVC).
# CFLAGS_DEBUG may be set to "-Zi -Od", "-DDEBUGIT" "-Zi -Od -DDEBUGIT" or left
blank
CC=cl.exe
CFLAGS_OS=-DSQLWINT -MT -DWIN32 -J -Zp8 -DREG_KIT_METHOD
CFLAGS_OUT=/Fo
CFLAGS_DEBUG=

# Linker Configuration (MSVC)
LD_EXEC=link.exe
LD_STORP=link.exe
LDFLAGS_EXEC=
LDFLAGS_SHLIB=/DLL
LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpcc.def
LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib /LIBPATH:"C:\Program Files\Microsoft
Visual Studio\VC98\Lib" db2api.lib winmm.lib
LDFLAGS_OUT=/OUT:

# Library Configuration
AR=lib.exe
ARFLAGS=
ARFLAGS_LIB=
ARFLAGS_OUT=/OUT:

# OS Commands
ERASE=del /F
ERASEDIR=rmdir /S
MOVE=MOVE
COPY=COPY

# OS File Extensions & Path Separator
OBJEXT=.obj
LIBEXT=.lib
SHLIBEXT=.dll
BINEXT=.exe
SLASH=\\
CMDSEP=&
```

Src.Cli/Makefile

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
# ====== =========== === ========================================

#
# Makefile - Makefile for Src.Cli (RTE/Driver Interface)
#

!include $(TPCC_ROOT)/Makefile.config

#
################################################################################
# Preprocessor Compiler and Linker Flags
#
################################################################################

BND_OPTS =    GRANT PUBLIC \
        MESSAGES $*.bnd.msg
PRP_OPTS =    BINDFILE \
        ISOLATION RR \
        EXPLAIN ALL \
        MESSAGES $*.prep.msg \
        LEVEL $(TPCC_VERSION) \
        NOLINEMACRO

INCLUDES =-I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =    $(CFLAGS_OS) $(INCLUDES) $(CFLAGS_DEBUG) \
        $(UOPTS) -D$(DB2EDITION) -D$(DB2VERSION) -D$(TPCC_SPTYPE)

OBJS =     $(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT) \
        $(TPCC_ROOT)/Src.Common/tpccctx$(OBJEXT) \
        tpcccli$(OBJEXT)

LIBS =      tpcccli$(LIBEXT)

#
################################################################################
# User Targets
#
################################################################################

all:    connect $(OBJS) plan $(LIBS) disconnect
    $(AR) $(ARFLAGS) $(ARFLAGS_OUT)tpcccli$(LIBEXT) $(OBJS) $(ARFLAGS_LIB)
    @echo "------------------------------------------------------------"
    @echo "Please copy lval.h, db2tpcc.h, and tpcccli$(LIBEXT) to"
    @echo "a place where they can be #included and linked with the"
    @echo "RTE/driver code."
    @echo "------------------------------------------------------------"

clean:
    - $(ERASE) *.msg *.bnd *.plan *$(OBJEXT) *$(LIBEXT) tpcccli.c

#
################################################################################
# Helper Targets
#
################################################################################

connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

plan:
    - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s $(TPCC_SCHEMA) -w -1 -n
TPCCCLI -g -# 0 -o TPCCCLI.exfmt.plan
    - db2expln -d $(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p TPCCCLI -s 0 -g -o
TPCCCLI.expln.plan

rebind:  connect
    db2 bind tpcccli.bnd $(BND_OPTS) QUERYOPT 7

#
################################################################################
# Build Rules
#
################################################################################

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

tpcccli.c:
    @echo "Prepping $*.sqc"
    -db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
    @echo "Binding $*.bnd"
    db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7

#
################################################################################
# Dependencies
#
################################################################################

# Client Library:
tpcccli$(LIBEXT):  $(OBJS)

# Source
tpcccli$(OBJEXT): tpcccli.c

# Headers
tpcccli.c:       $(TPCC_ROOT)/include/db2tpcc.h $(TPCC_ROOT)/include/lval.h
```

Src.Cli/tpcccli.sqc

```
/*********************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*********************************************************************************/

/*
 * tpcccli.sqc - Client/Server code for TPCC
 */

#include <stdlib.h>
#include <errno.h>
#include "db2tpcc.h"
#include "tpccapp.h"
#include "tpccdbg.h"

#include "sqlca.h"
#include "sql.h"

// ---------------------------------------------------------------
// New Order CLIENT
// ---------------------------------------------------------------

static int itemComparison ( const void * a , const void * b )
{
  struct in_items_struct * one = (struct in_items_struct *) a ;
  struct in_items_struct * two = (struct in_items_struct *) b ;


  if ( one->s_OL_I_ID != two->s_OL_I_ID )
  {
    return (  one->s_OL_I_ID - two->s_OL_I_ID  ) ;
  }
  else
  {
    return (  one->s_OL_SUPPLY_W_ID - two->s_OL_SUPPLY_W_ID  ) ;
  }
}

int neword_sql (  struct in_neword_struct  * in_neword
                , struct out_neword_struct * neword      )
{
  struct sqlca sqlca ;

  EXEC SQL BEGIN DECLARE SECTION;

    struct vc_new_in
    {
      short len;
      char data[ 262 ] ;
    } * pHostvarInput ;

    struct vc_new_out
    {
      short len;
      char data[ 682 ] ;
    } * pHostvarOutput ;

  EXEC SQL END DECLARE SECTION;

  int clientRc = TRAN_OK ;

  int itemIndex = 0 ;


  in_neword->s_all_local  = 1 ;

  for ( itemIndex = 0 ;
        itemIndex < 15 && in_neword->in_item[ itemIndex ].s_OL_I_ID !=
UNUSED_ITEM_ID ;
        itemIndex++
      )
  {
    if ( in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID != in_neword->s_W_ID )
    {
      in_neword->s_all_local = 0 ;
    }
  }

  in_neword->s_O_OL_CNT = itemIndex ;

  qsort( in_neword->in_item, in_neword->s_O_OL_CNT
       ,  sizeof ( in_neword->in_item[ 0 ] )
       , itemComparison
       ) ;

  pHostvarInput      = (struct vc_new_in *) in_neword ;
  pHostvarInput->len = sizeof(struct in_neword_struct) - SPGENERAL_ADJUST ;

  pHostvarOutput     = (struct vc_new_out *) neword;
  pHostvarOutput->len = sizeof(struct out_neword_struct) - SPGENERAL_ADJUST ;

#ifdef DEBUGIT
  new_debug(neword, in_neword, "Client before SP call");
#endif /* DEBUGIT */

#ifdef SWAP_ENDIAN
  for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT; itemIndex++)
  {
    SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_I_ID);
    SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
    SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_QUANTITY);
  }
  SWAP_BYTE(in_neword->s_C_ID);
  SWAP_BYTE(in_neword->s_W_ID);
  SWAP_BYTE(in_neword->s_D_ID);
  SWAP_BYTE(in_neword->s_O_OL_CNT);
  SWAP_BYTE(in_neword->s_all_local);
  SWAP_BYTE(in_neword->duplicate_items);
#endif  //SWAP_ENDIAN

  EXEC SQL CALL news ( :*pHostvarInput, :*pHostvarOutput );

#ifdef SWAP_ENDIAN
  SWAP_BYTE(in_neword->s_C_ID);
  SWAP_BYTE(in_neword->s_W_ID);
  SWAP_BYTE(in_neword->s_D_ID);
  SWAP_BYTE(in_neword->s_O_OL_CNT);
  SWAP_BYTE(in_neword->s_all_local);
  SWAP_BYTE(in_neword->duplicate_items);
  for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT; itemIndex++)
  {
    SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_I_ID);
    SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID);
    SWAP_BYTE(in_neword->in_item[ itemIndex ].s_OL_QUANTITY);
  }

  SWAP_BYTE(neword->s_W_TAX);
  SWAP_BYTE(neword->s_D_TAX);
  SWAP_BYTE(neword->s_C_DISCOUNT);
  SWAP_BYTE(neword->s_total_amount);
  SWAP_BYTE(neword->s_O_ID);
  SWAP_BYTE(neword->s_O_OL_CNT);
  SWAP_BYTE(neword->s_transtatus);
  SWAP_BYTE(neword->deadlocks);
  for (itemIndex=0; itemIndex<in_neword->s_O_OL_CNT; itemIndex++)
  {
    SWAP_BYTE(neword->item[ itemIndex ].s_I_PRICE);
    SWAP_BYTE(neword->item[ itemIndex ].s_OL_AMOUNT);
    SWAP_BYTE(neword->item[ itemIndex ].s_S_QUANTITY);
  }
#endif  //SWAP_ENDIAN

  if ( sqlca.sqlcode == 0 )
  {
    float wtax = neword->s_W_TAX ;
    float dtax = neword->s_D_TAX ;
    float cdisc = neword->s_C_DISCOUNT ;
    float factor = (1.0 - cdisc) * (1.0 + wtax + dtax) ;

    // Compute order total

    neword->s_total_amount = 0 ;

    for ( itemIndex = 0 ;
          itemIndex < in_neword->s_O_OL_CNT ; // from input , not output
          itemIndex++
        )
    {
      if ( neword->item[ itemIndex ].s_I_PRICE > 0 )  // A zero price signifies a bad item
      {
        neword->item[ itemIndex ].s_OL_AMOUNT =  neword-
>item[ itemIndex ].s_I_PRICE *
                                                 in_neword->in_item[ itemIndex ].s_OL_QUANTITY ;  //
reference input value

        neword->s_total_amount += neword->item[ itemIndex ].s_OL_AMOUNT ;

      }
    }

    neword->s_total_amount *= factor;
  }
  else
  {
    sqlerror( NEWORD_SQL, "NEW", __FILE__, __LINE__, &sqlca) ;
    neword->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;
  }

#ifdef DEBUGIT
  new_debug(neword, in_neword, "Client after SP call");
#endif /* DEBUGIT */

  if (neword->s_transtatus <= FATAL_SQLERROR)
  {
    new_debug(neword, in_neword, "NEW failed");
    clientRc = FATAL_SQLERROR ;
  }

  if (neword->s_transtatus == INVALID_ITEM)
  {
    clientRc = INVALID_ITEM ;
  }

  return ( clientRc ) ;
}

// ---------------------------------------------------------------
// Payment CLIENT
// ---------------------------------------------------------------

int payment_sql (   struct in_payment_struct * in_payment
                  , struct out_payment_struct * payment     )
```

```
{
  struct sqlca sqlca ;

  int clientRc = TRAN_OK ;

  EXEC SQL BEGIN DECLARE SECTION;

    // Inputs

    float h_amount ;
    sqlint32 in_c_id ;

    struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;

    sqlint32 w_id ;
    sqlint32 c_w_id ;
    short    d_id ;
    short    c_d_id ;

    // Outputs

    sqlint32 c_id ;

    double  c_credit_lim ;
    float   c_discount ;
    double  c_balance ;

    char    w_street_1 [ 20 ] , w_street_2 [ 20 ] ;
    char    w_city [ 20 ] ,     w_state [ 2 ] ,      w_zip [ 9 ] ;

    char    d_street_1 [ 20 ] , d_street_2 [ 20 ] ,  d_city [ 20 ] ;
    char    d_state [ 2 ] ,     d_zip [ 9 ] ,        c_first [ 16 ] ;

    char    c_last [ 16 ] ;

    char    c_middle [ 2 ]    , c_street_1 [ 20 ] ;
    char    c_street_2 [ 20 ] , c_city [ 20 ] ,      c_state [ 2 ] ;
    char    c_zip [ 9 ] ,       c_phone [ 16 ] ;

    char    c_credit [ 2 ] ;

    char    c_since [27];

    char    c_data [ 200 ] ;
    short   c_data_indicator = 0 ;

    char    h_date [27];

    struct c_data_prefix_c_last_type { short len ; char data[ 28 ] ; } c_data_prefix_c_last ;
    struct c_data_prefix_c_id_type   { short len ; char data[ 34 ] ; } c_data_prefix_c_id ;


  EXEC SQL END DECLARE SECTION;


  // Input redirects

  #define h_amount          in_payment->s_H_AMOUNT
  #define in_c_id           in_payment->s_C_ID

  #define w_id              in_payment->s_W_ID
  #define d_id              in_payment->s_D_ID

  #define c_d_id            in_payment->s_C_D_ID
  #define c_w_id            in_payment->s_C_W_ID

  // Output redirects

  #define c_credit_lim      payment->s_C_CREDIT_LIM
  #define c_discount        payment->s_C_DISCOUNT
  #define c_balance         payment->s_C_BALANCE

  #define c_id              payment->s_C_ID
  #define c_last            payment->s_C_LAST

  #define c_first           payment->s_C_FIRST
  #define c_middle          payment->s_C_MIDDLE
  #define c_street_1        payment->s_C_STREET_1
  #define c_street_2        payment->s_C_STREET_2
  #define c_city            payment->s_C_CITY
  #define c_state           payment->s_C_STATE
  #define c_zip             payment->s_C_ZIP
  #define c_phone           payment->s_C_PHONE
  #define c_credit          payment->s_C_CREDIT
  #define c_since           payment->s_C_SINCE_time
  #define c_data            payment->s_C_DATA

  #define w_street_1        payment->s_W_STREET_1
  #define w_street_2        payment->s_W_STREET_2
  #define w_city            payment->s_W_CITY
  #define w_state           payment->s_W_STATE
  #define w_zip             payment->s_W_ZIP

  #define d_street_1        payment->s_D_STREET_1
  #define d_street_2        payment->s_D_STREET_2
  #define d_city            payment->s_D_CITY
  #define d_state           payment->s_D_STATE
  #define d_zip             payment->s_D_ZIP

  #define h_date            payment->s_H_DATE_time

  payment->deadlocks = -1 ;
  payment->s_transtatus = TRAN_OK ;

  if (c_w_id == 0) { c_w_id = w_id; }
  if (c_d_id == 0) { c_d_id = d_id; }

#ifdef DEBUGIT
  pay_debug(payment, in_payment, "Client before SQL call");
#endif /* DEBUGIT */

  // Create c_data_prefix strings and copy some elements from
  // in -> out struct outside of retry_tran loop

  if ( in_c_id == 0 )
  {

    c_data_prefix_c_last.len = sprintf( c_data_prefix_c_last.data, " %2.2d %6.6d %2.2d
%6.6d %06.2f",  c_d_id , c_w_id , d_id , w_id , h_amount ) ;

    // Setup the input c_last varchar
    c_last_input.len = strlen( in_payment->s_C_LAST ) ;
    memcpy( c_last_input.data , in_payment->s_C_LAST , c_last_input.len ) ;

    // Copy to the output structure
    memcpy( payment->s_C_LAST , in_payment->s_C_LAST, sizeof( payment-
>s_C_LAST ) ) ;

  } else {

    // Copy c_id to the output structure
    c_id = in_c_id ;


    c_data_prefix_c_id.len = sprintf( c_data_prefix_c_id.data, " %5.5d %2.2d %6.6d %2.2d
%6.6d %06.2f", c_id , c_d_id , c_w_id , d_id , w_id , h_amount) ;

  }

retry_tran:

  payment->deadlocks ++ ;

  if ( in_c_id == 0 )
  {
      EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

        SELECT   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
             , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
             , C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
             , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT,
C_CREDIT_LIM
             , C_DISCOUNT, C_BALANCE, C_DATA, H_DATE

          INTO   :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
             , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
             , :c_id , :c_first , :c_middle ,   :c_street_1 , :c_street_2 , :c_city , :c_state
             , :c_zip , :c_phone , :c_since , :c_credit , :c_credit_lim
             , :c_discount , :c_balance, :c_data :c_data_indicator, :h_date
          FROM TABLE ( PAY_C_LAST(   :w_id
                                   , :d_id
                                   , :c_w_id
                                   , :c_d_id
                                   , :c_last_input
                                   , CAST(:h_amount AS DECIMAL(6,2))
                                   , :c_data_prefix_c_last
                                  )

                 ) AS T (   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
                          , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
                          , C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
                          , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT,
C_CREDIT_LIM
                          , C_DISCOUNT, C_BALANCE, C_DATA, H_DATE
                         )
        ;

        COMMIT ;

      END COMPOUND ;

  }
  else
  {
      EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

        SELECT   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
             , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
             , C_LAST, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
             , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT,
C_CREDIT_LIM
             , C_DISCOUNT, C_BALANCE, C_DATA, H_DATE

          INTO   :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
             , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
             , :c_last, :c_first , :c_middle , :c_street_1 , :c_street_2 , :c_city , :c_state
             , :c_zip , :c_phone , :c_since , :c_credit , :c_credit_lim
             , :c_discount , :c_balance, :c_data :c_data_indicator, :h_date
          FROM TABLE ( PAY_C_ID(   :w_id
                                   , :d_id
                                   , :c_w_id
                                   , :c_d_id
                                   , :in_c_id
                                   , CAST(:h_amount AS DECIMAL(6,2))
                                   , :c_data_prefix_c_id
                                  )

                 ) AS T(   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
                         , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
                         , C_LAST, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
```

```
                        , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT,
C_CREDIT_LIM
                        , C_DISCOUNT, C_BALANCE, C_DATA, H_DATE
                       )
        ;

        COMMIT ;

      END COMPOUND ;

  }

#ifdef DEBUGIT
   pay_debug(payment, in_payment, "Client after SQL call");
#endif /* DEBUGIT */

  if ( sqlca.sqlcode != 0 )
  {
    DLCHK( retry_tran ) ;

    sqlerror( PAYMENT_SQL , "PAY" , __FILE__, __LINE__ , &sqlca) ;
    payment->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;

    pay_debug( payment, in_payment, "PAY failed" ) ;

    EXEC SQL ROLLBACK WORK ;

    if ( sqlca.sqlcode != 0 )
    {
      sqlerror( PAYMENT_SQL, "ROLLBACK FAILED", __FILE__, __LINE__, &sqlca ) ;
    }
  }

  return ( clientRc ) ;
}

// ---------------------------------------------------------------------
// Order Status CLIENT
// ---------------------------------------------------------------------

int ordstat_sql  (  struct in_ordstat_struct * in_ordstat
              , struct out_ordstat_struct * ordstat)
{
  struct sqlca sqlca ;

  EXEC SQL BEGIN DECLARE SECTION;

    struct vc_ord_in
    {
      short len ;
      char data[ 42 ] ;
    } * in_ord ;

    struct vc_ord_out
    {
      short len ;
      char data[ 822 ] ;
    } * out_ord ;

  EXEC SQL END DECLARE SECTION;

  int clientRc = TRAN_OK ;
  int itemIndex = 0 ;

  in_ord      = (struct vc_ord_in *) in_ordstat ;
  in_ord->len = sizeof(struct in_ordstat_struct) - SPGENERAL_ADJUST ;

  out_ord      = (struct vc_ord_out *) ordstat ;
  out_ord->len = sizeof(struct out_ordstat_struct) - SPGENERAL_ADJUST ;


#ifdef DEBUGIT
   ord_debug(ordstat, in_ordstat, "Client before SP call");
#endif /* DEBUGIT */

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_ordstat->s_C_ID);
   SWAP_BYTE(in_ordstat->s_W_ID);
   SWAP_BYTE(in_ordstat->s_D_ID);
#endif  //SWAP_ENDIAN

   EXEC SQL CALL ords ( :*in_ord, :*out_ord ) ;

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_ordstat->s_C_ID);
   SWAP_BYTE(in_ordstat->s_W_ID);
   SWAP_BYTE(in_ordstat->s_D_ID);

   SWAP_BYTE(ordstat->s_C_BALANCE);
   SWAP_BYTE(ordstat->s_C_ID);
   SWAP_BYTE(ordstat->s_O_ID);
   SWAP_BYTE(ordstat->s_O_CARRIER_ID);
   SWAP_BYTE(ordstat->s_ol_cnt);
   SWAP_BYTE(ordstat->s_transtatus);
   SWAP_BYTE(ordstat->deadlocks);
   for (itemIndex=0; itemIndex<ordstat->s_ol_cnt; itemIndex++)
   {
     SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_AMOUNT);
     SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_I_ID);
     SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_SUPPLY_W_ID);
     SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_QUANTITY);
   }
#endif  //SWAP_ENDIAN

   if ( sqlca.sqlcode == 0 )
   {
     // Propogate the field we already knew into the output structure
     // 60% of the time, we already new c_last (input c_id is 0)

     if ( in_ordstat->s_C_ID == 0 )
     {
       memcpy( ordstat->s_C_LAST , in_ordstat->s_C_LAST, sizeof( ordstat-
>s_C_LAST ) ) ;
     }
     else
     {
       ordstat->s_C_ID = in_ordstat->s_C_ID ;
     }
   }
   else
   {
     sqlerror( ORDSTAT_SQL, "ORD", __FILE__, __LINE__, &sqlca) ;
     ordstat->s_transtatus = FATAL_SQLERROR ;
     clientRc = FATAL_SQLERROR ;
   }

#ifdef DEBUGIT
   ord_debug(ordstat, in_ordstat, "Client after SP call");
#endif /* DEBUGIT */

   if ( ordstat->s_transtatus <= FATAL_SQLERROR )
   {
     ord_debug(ordstat, in_ordstat, "ORD failed");
     clientRc = FATAL_SQLERROR ;
   }

   return ( clientRc ) ;
}

// ---------------------------------------------------------------------


// Delivery CLIENT
// ---------------------------------------------------------------------

int delivery_sql (   struct in_delivery_struct  * in_delivery
                , struct out_delivery_struct * delivery      )
{
  struct sqlca sqlca ;

  EXEC SQL BEGIN DECLARE SECTION;

    struct vc_del_in
    {
      short len ;
      char data[ 14 ] ;
    } * in_del ;

    struct vc_del_out
    {
      short len;
      char data[ 50 ] ;
    } * out_del ;

  EXEC SQL END DECLARE SECTION;

  int clientRc = TRAN_OK ;
  int orderIndex = 0 ;

  in_del      = (struct vc_del_in *) in_delivery ;
  in_del->len = sizeof(struct in_delivery_struct) - SPGENERAL_ADJUST;

  out_del      = (struct vc_del_out *) delivery ;
  out_del->len = sizeof(struct out_delivery_struct) - SPGENERAL_ADJUST;

#ifdef DEBUGIT
   del_debug(delivery, in_delivery, "Client before SP call");
#endif /* DEBUGIT */

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_delivery->s_W_ID);
   SWAP_BYTE(in_delivery->s_O_CARRIER_ID);
#endif  //SWAP_ENDIAN

   EXEC SQL CALL dels ( :*in_del, :*out_del ) ;

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_delivery->s_W_ID);
   SWAP_BYTE(in_delivery->s_O_CARRIER_ID);

   for (orderIndex=0; orderIndex<10; orderIndex++) {
     SWAP_BYTE(delivery->s_O_ID[ orderIndex ]);
   }
   SWAP_BYTE(delivery->s_transtatus);
   SWAP_BYTE(delivery->deadlocks);
#endif  //SWAP_ENDIAN

#ifdef DEBUGIT
   del_debug(delivery, in_delivery, "Client after SP call");
#endif /* DEBUGIT */

   if ( sqlca.sqlcode != 0 )
   {
     sqlerror( DELIVERY_SQL, "DEL", __FILE__, __LINE__, &sqlca) ;
     delivery->s_transtatus = FATAL_SQLERROR ;
     clientRc = FATAL_SQLERROR ;
   }

   if ( delivery->s_transtatus <= FATAL_SQLERROR )
   {
     del_debug(delivery, in_delivery, "DEL failed");
     clientRc = FATAL_SQLERROR ;
```

```
  }

  return ( clientRc ) ;
}

// ---------------------------------------------------------------------
// Stock CLIENT
// ---------------------------------------------------------------------

#undef w_id
#undef d_id

int stocklev_sql  (   struct in_stocklev_struct  * in_stocklev
                , struct out_stocklev_struct * stocklev     )
{
  struct sqlca sqlca ;

  int clientRc = TRAN_OK ;

  EXEC SQL BEGIN DECLARE SECTION;

    // input

    sqlint32       threshold ;

    // output

    sqlint32    low_stock ;

  EXEC SQL END DECLARE SECTION;

  #define w_id      in_stocklev->s_W_ID
  #define d_id      in_stocklev->s_D_ID
  #define threshold in_stocklev->s_threshold
  #define low_stock stocklev->s_low_stock

  stocklev->deadlocks = -1 ;
  stocklev->s_transtatus = TRAN_OK ;

#ifdef DEBUGIT
  stk_debug(stocklev, in_stocklev, "Client before SQL call");
#endif /* DEBUGIT */

retry_tran:

  stocklev->deadlocks ++ ;

  EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

    SELECT COUNT( S_I_ID ) INTO :low_stock

      FROM ( SELECT DISTINCT S_I_ID

               FROM ORDER_LINE , STOCK , DISTRICT

              WHERE D_W_ID = :w_id
                AND D_ID = :d_id
                AND OL_O_ID < d_next_o_id
                AND OL_O_ID >= ( d_next_o_id - 20 )
                AND OL_W_ID = D_W_ID
                AND OL_D_ID = D_ID
                AND S_I_ID = OL_I_ID
                AND S_W_ID = OL_W_ID
                AND S_QUANTITY < :threshold

           ) OLS

      WITH CS
    ;

    COMMIT ;

  END COMPOUND ;

#ifdef DEBUGIT
  stk_debug(stocklev, in_stocklev, "Client after SQL call");
#endif /* DEBUGIT */

  if ( sqlca.sqlcode != 0 )
  {
    DLCHK( retry_tran ) ;

    sqlerror( STOCKLEV_SQL , "STK" , __FILE__, __LINE__ , &sqlca);
    stocklev->s_transtatus = FATAL_SQLERROR ;
    clientRc = FATAL_SQLERROR ;

    stk_debug( stocklev, in_stocklev, "STK failed" ) ;

    EXEC SQL ROLLBACK WORK ;

    if ( sqlca.sqlcode != 0 )
    {
      sqlerror( STOCKLEV_SQL, "ROLLBACK FAILED", __FILE__, __LINE__, &sqlca ) ;
    }
  }

  return ( clientRc ) ;
}
```

Src.Common/Makefile

```
#################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#################################################################################
######

#
# Makefile - Makefile for Src.Common
#

!include $(TPCC_ROOT)/Makefile.config

#
#################################################################################
# Preprocessor, Compiler and LInker Flags
#
#################################################################################

BND_OPTS =    GRANT PUBLIC \
        MESSAGES $*.bnd.msg
PRP_OPTS =    BINDFILE \
        OPTLEVEL 1 \
        ISOLATION RR \
        MESSAGES $*.prep.msg \
        LEVEL $(TPCC_VERSION) \
        NOLINEMACRO

INCLUDES =-I$(TPCC_SQLLIB)$(SLASH)include -I$(TPCC_ROOT)$(SLASH)include

CFLAGS =    $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDES) \
        -DSQLA_NOLINES -D$(DB2EDITION) -D$(DB2VERSION) \
        -D$(TPCC_SPTYPE)

UTIL_OBJ =  tpccmisc$(OBJEXT) tpccdbg$(OBJEXT)
UTIL_OBJ_DB2 = tpccctx$(OBJEXT)

#
#################################################################################
# User Targets
#
#################################################################################

all:    dbgen connect $(UTIL_OBJ_DB2) disconnect

dbgen:  $(UTIL_OBJ)

clean:
    - $(ERASE) *$(OBJEXT) *.bnd *.msg tpccctx.c

#
#################################################################################
# Helper Targets
#
#################################################################################

connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

rebind:  connect
    db2 bind tpccctx.bnd $(BND_OPTS)

#
#################################################################################
# Build Rules
#
#################################################################################

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
    @echo "Prepping $*.sqc"
    -db2 prep $*.sqc $(PRP_OPTS)
    @echo "Binding $*.bnd"
    db2 bind $*.bnd $(BND_OPTS)

#
#################################################################################
# Dependencies
#
#################################################################################

# Source
tpccdbg$(OBJEXT):   tpccdbg.c
tpccctx$(OBJEXT):    tpccctx.c
tpccmisc$(OBJEXT): tpccmisc.c

# Headers
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Common/tpccctx.sqc

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
```

```
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*******************************************************************************/

/*
 *
 * tpccctx.sqc - TPCC context code
 *
 */

#include <string.h>
#include <sqlutil.h>
#include "db2tpcc.h"
#include "tpccdbg.h"

int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username, char *in_password);
int disconnect_from_TM(void);

int connect_to_TM(char *in_dbname)
{
  return connect_to_TM_auth(in_dbname, "", "");
}

int connect_to_TM_auth(char *in_dbname, char *in_username, char *in_password)
{
   SQL_STRUCTURE sqlca  sqlca;
   int ConnectSQLCODE = 0;

   EXEC SQL BEGIN DECLARE SECTION;
   char dbname[9];
   char username[129];
   char password[15];
   EXEC SQL END DECLARE SECTION;

   /* Copy 9 characters - 8 for dbname, 1 for NULL */
   strncpy(dbname,in_dbname,9);
   if (strcmp(in_username,"") == 0)
   {
      EXEC SQL CONNECT TO :dbname IN SHARE MODE;
   } else {
      strncpy(username,in_username,128);
      strncpy(password,in_password,14);
      EXEC SQL CONNECT TO :dbname IN SHARE MODE USER :username
USING :password;
   }

   ConnectSQLCODE = SQLCODE;
   if (ConnectSQLCODE != 0)
   {
      sqlerror( CLIENT_SQL, "CONNECT", __FILE__, __LINE__, &sqlca);

      return ConnectSQLCODE;
   }

   return 0;
}

int disconnect_from_TM(void)
{
   SQL_STRUCTURE sqlca  sqlca;
   int DisconnectSQLCODE = 0;

   EXEC SQL CONNECT RESET;

   DisconnectSQLCODE = SQLCODE;
   if (DisconnectSQLCODE != 0) {
      sqlerror( CLIENT_SQL, "DISCONNECT", __FILE__, __LINE__, &sqlca);
   }

   if (DisconnectSQLCODE) {
      return DisconnectSQLCODE;
   }
   return 0;
}
```

Src.Common/tpccdbg.c

```
/*******************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*******************************************************************************/

/*
 * tccdbg.c - Debugging Routines
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>

#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"

#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128

void  del_print();
void  new_print();
void  ord_print();
void  pay_print();
void  stk_print();

void current_tmstmp(char *buf);

static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";

/*--------------------------------------------------------------------*/
/*       InitializeDebug                                              */
/*--------------------------------------------------------------------*/
__inline void InitializeDebug(void) {
   if (debugInit == 0) {
      char *p = getenv("TPCC_DEBUGDIR");
      if (p) {
         strncpy(debugPath, p, DEBUG_PATH_SIZE);
      } else {
         strcpy(debugPath, "C:\\temp");
      }
      strcat(debugPath, "\\");
   }
   debugInit = 1;
}

/*--------------------------------------------------------------------*/
/*       sqlerror                                                     */
/*--------------------------------------------------------------------*/
void sqlerror(int tranType, char *msg, char *file, int line, SQL_STRUCTURE sqlca *psqlca)
{
   FILE *err_fp = NULL;
   char  err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
   char  tranName[16];
   int j,k;
   char  timeStamp[27];
   char  errStr[512] = "";

   InitializeDebug();
   strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
   current_tmstmp(&timeStamp[0]);
   timeStamp[19] = (char)NULL;

   switch(tranType)
   {
      case NEWORD_SQL:
         // sprintf(err_fn, "%d.err.out", getpid());
         strcat(err_fn, "new.err.out");
         strcpy(tranName, "NEW_ORDER");
         break;

      case DELIVERY_SQL:
         // sprintf(err_fn, "%d.err.out", getpid());
         strcat(err_fn, "del.err.out");
         strcpy(tranName, "DELIVERY");
         break;

      case PAYMENT_SQL:
         // sprintf(err_fn, "%d.err.out", getpid());
         strcat(err_fn, "pay.err.out");
         strcpy(tranName, "PAYMENT");
         break;

      case ORDSTAT_SQL:
         // sprintf(err_fn, "%d.err.out", getpid());
         strcat(err_fn, "ord.err.out");
         strcpy(tranName, "ORDER_STAT");
         break;

      case STOCKLEV_SQL:
         //sprintf(err_fn, "%d.err.out", getpid());
         strcat(err_fn, "stk.err.out");
         strcpy(tranName, "STOCK_LVL");
         break;

      case 0:
         strcat(err_fn, "cli.err.out");
         strcpy(tranName, "CLIENT");
         break;

      default:
         return;

   }

   /* Generate Formatted Error Message */
   sqlaintp(errStr, 512, 78, psqlca);

   if ((err_fp = fopen(err_fn, "a+")) == NULL)
   {
      return;
   }
```

```
  fprintf(err_fp, "----------------------------------------\n");
  fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
  fprintf(err_fp, "FILE %s (%u)\n", file, line);
  fprintf(err_fp, "SQLCODE %d ", psqlca->sqlcode);
  fprintf(err_fp, "PID %d ", getpid());
  fprintf(err_fp, "TIME %s\n", timeStamp);
  fprintf(err_fp, "----------------------------------------\n");
  fprintf(err_fp, "%s", errStr);
  fprintf(err_fp, "----------------------------------------\n");

  if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
  {
    fprintf(err_fp, "slerrmc:  ");

    for(j = 0; j < 5; j++)
    {
      for(k = 0; k < 16; k++) {
        int pos = j * 16 + k;
        if (pos < 70) fprintf(err_fp, "%02x ", psqlca->sqlerrmc[pos]);
        else fprintf(err_fp, "   ");
      }
      fprintf(err_fp,"   |");
      for(k = 0; k < 16; k++) {
        int pos = j * 16 + k;
        char c = ' ';
        if (pos < 70) {
          c = psqlca->sqlerrmc[pos];
          if (!isprint(c)) c = ' ';
        }
        fprintf(err_fp,"%c", c);
      }
      fprintf(err_fp,"|\n");
      if (j < 4) fprintf(err_fp,"          ");
    }
  }

  fprintf(err_fp, "sqlerrp:  ");
  for(j = 0; j < 8; j++)
    fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
  fprintf(err_fp,"\n");

  fprintf(err_fp, "sqlerrd: ");
  for(j = 0; j < 6; j++)
    fprintf(err_fp, " %d", psqlca->sqlerrd[j]);
  fprintf(err_fp,"\n");

  if (psqlca->sqlwarn[0] != ' ')
  {
    fprintf(err_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
      fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
    fprintf(err_fp,"\n");
  }

  fprintf(err_fp, "\n");

  fclose(err_fp);
}


/*-------------------------------------------------------------*/
/* del_debug                                                   */
/*-------------------------------------------------------------*/
void del_debug (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct  *in_delivery,
                char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "del.debug.out");
  del_print(delivery_ptr, in_delivery, debug_fn, msg);
}

/*-------------------------------------------------------------*/
/* del_print                                                   */
/*-------------------------------------------------------------*/
void del_print (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct  *in_delivery,
                char *filename,
                char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];
  int   j;

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Delivery debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==================================================\n");

  fprintf(debug_fp,"in_delivery_struct {\n");
  fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
          in_delivery->s_W_ID, in_delivery->s_W_ID);
  fprintf(debug_fp,"\ts_O_CARRIER_ID = %d (%X)\n",
          in_delivery->s_O_CARRIER_ID, in_delivery->s_O_CARRIER_ID);
  fprintf(debug_fp,"}\n\n");

  fprintf(debug_fp,"out_delivery_struct {\n");
  fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
          delivery_ptr->s_transtatus,delivery_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
          delivery_ptr->deadlocks,delivery_ptr->deadlocks);

  for (j = 0; j < 10; j++) {
    fprintf(debug_fp,"\t\ts_O_ID[%d]         = %d\n",
            j, delivery_ptr->s_O_ID[j]);
  }
  fprintf(debug_fp,"\t}\n}\n\n");
  fclose(debug_fp);
}


/*-------------------------------------------------------------*/
/* new_debug                                                   */
/*-------------------------------------------------------------*/
void new_debug (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "new.debug.out");
  new_print(neword_ptr, in_neword, debug_fn, msg);
}

/*-------------------------------------------------------------*/
/* new_print                                                   */
/*-------------------------------------------------------------*/
void new_print (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *filename,
                char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];
  int   j, items;

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"New order debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==================================================\n");

  fprintf(debug_fp,"in_neword_struct {\n");

  fprintf(debug_fp,"\ts_C_ID         = %d (%X)\n",
          in_neword->s_C_ID, in_neword->s_C_ID);
  fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
          in_neword->s_W_ID, in_neword->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID         = %d (%X)\n",
          in_neword->s_D_ID, in_neword->s_D_ID);
  fprintf(debug_fp,"\ts_O_OL_CNT     = %d (%X)\n",
          in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
  fprintf(debug_fp,"\ts_all_local    = %d (%X)\n",
          in_neword->s_all_local, in_neword->s_all_local);
// fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
//         in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp,"\tduplicate_items= %d (%X)\n",
//         in_neword->duplicate_items, in_neword->duplicate_items);

  fprintf(debug_fp,"\titems {\n");
  items = in_neword->s_O_OL_CNT;
  for (j=0; j<items; j++) {
    if(j != 0)
      fprintf(debug_fp,"\n");
    fprintf(debug_fp,"\t\ts_OL_I_ID[%d]        = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_I_ID, in_neword->in_item[j].s_OL_I_ID);
    fprintf(debug_fp,"\t\ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_SUPPLY_W_ID, in_neword-
>in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp,"\t\ts_OL_QUANTITY[%d]    = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_QUANTITY, in_neword-
>in_item[j].s_OL_QUANTITY);
  }
  fprintf(debug_fp,"\t}\n}\n\n");

  fprintf(debug_fp,"out_neword_struct {\n");
  fprintf(debug_fp,"\ts_C_LAST       = %s\n",
          neword_ptr->s_C_LAST);
  fprintf(debug_fp,"\ts_C_CREDIT     = %s\n",
          neword_ptr->s_C_CREDIT);
  fprintf(debug_fp,"\ts_W_TAX        = %04.4f \n",
          neword_ptr->s_W_TAX);
  fprintf(debug_fp,"\ts_D_TAX        = %04.4f \n",
          neword_ptr->s_D_TAX);
  fprintf(debug_fp,"\ts_C_DISCOUNT   = %04.4f \n",
          neword_ptr->s_C_DISCOUNT);
  fprintf(debug_fp,"\ts_O_ID         = %d (%X)\n",
          neword_ptr->s_O_ID, neword_ptr->s_O_ID);
  fprintf(debug_fp,"\ts_O_OL_CNT     = %d (%X)\n",
          neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
  fprintf(debug_fp,"\ts_O_ENTRY_D    = %s \n",
```

```
        neword_ptr->s_O_ENTRY_D_time);
   fprintf(debug_fp,"\ts_total_amount = %.2f \n",
        neword_ptr->s_total_amount);
   fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
        neword_ptr->s_transtatus, neword_ptr->s_transtatus);
   fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
        neword_ptr->deadlocks, neword_ptr->deadlocks);

// fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
//         neword_ptr->s_W_ID, neword_ptr->s_W_ID);
// fprintf(debug_fp,"\ts_D_ID         = %d (%X)\n",
//         neword_ptr->s_D_ID, neword_ptr->s_D_ID);
// fprintf(debug_fp,"\ts_all_local    = %d (%X)\n",
//         neword_ptr->s_all_local, neword_ptr->s_all_local);
// fprintf(debug_fp,"\tduplicate_items= %d (%X)\n",
//         neword_ptr->duplicate_items, neword_ptr->duplicate_items);

   fprintf(debug_fp,"\titems {\n");
   items = neword_ptr->s_O_OL_CNT;
   for (j=0; j<items; j++) {
     if(j != 0)
       fprintf(debug_fp,"\n");
     fprintf(debug_fp,"\t\ts_I_NAME[%d]         = %s\n",
          j, neword_ptr->item[j].s_I_NAME);
     fprintf(debug_fp,"\t\ts_I_PRICE[%d]        = %.2f \n",
          j, neword_ptr->item[j].s_I_PRICE);
     fprintf(debug_fp,"\t\ts_OL_AMOUNT[%d]      = %.2f \n",
          j, neword_ptr->item[j].s_OL_AMOUNT);
     fprintf(debug_fp,"\t\ts_S_QUANTITY[%d]     = %d (%X)\n",
          j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr->item[j].s_S_QUANTITY);
     fprintf(debug_fp,"\t\ts_brand_generic[%d]  = %c\n",
          j, neword_ptr->item[j].s_brand_generic);
   }
   fprintf(debug_fp,"\t}\n}\n\n");
   fclose(debug_fp);
}



/*---------------------------------------------------------------------*/
/* ord_debug                                                           */
/*---------------------------------------------------------------------*/
void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                struct in_ordstat_struct  *in_ordstat,
                char *msg)
{
   char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

   InitializeDebug();
   strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
   strcat(debug_fn, "ord.debug.out");
   ord_print(ordstat_ptr, in_ordstat, debug_fn, msg);
}

/*---------------------------------------------------------------------*/
/* ord_print                                                           */
/*---------------------------------------------------------------------*/
void ord_print (struct out_ordstat_struct *ordstat_ptr,
                struct in_ordstat_struct  *in_ordstat,
                char *filename,
                char *msg)
{
   FILE *debug_fp;
   char  timeStamp[27];
   int   j, items;

   current_tmstmp(&timeStamp[0]);
   timeStamp[19] = (char)NULL;

   if ((debug_fp = fopen(filename, "a+")) == NULL)
   {
     return;
   }

   fprintf(debug_fp,"Order status debug information follows %s (%s)\n", timeStamp, msg);
   fprintf(debug_fp, " PID %d ", getpid());
   fprintf(debug_fp,"\n=================================================\n");

   fprintf(debug_fp,"in_ordstat_struct {\n");
   fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
          in_ordstat->s_W_ID, in_ordstat->s_W_ID);
   fprintf(debug_fp,"\ts_D_ID         = %d (%X)\n",
          in_ordstat->s_D_ID, in_ordstat->s_D_ID);
   fprintf(debug_fp,"\ts_C_ID         = %d (%X)\n",
          in_ordstat->s_C_ID, in_ordstat->s_C_ID);
   fprintf(debug_fp,"\ts_C_LAST       = %s\n",
          in_ordstat->s_C_LAST);
   fprintf(debug_fp,"}\n\n");

   fprintf(debug_fp,"out_ordstat_struct {\n");
   fprintf(debug_fp,"\ts_C_ID         = %d (%X)\n",
          ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);
   fprintf(debug_fp,"\ts_C_FIRST      = %s\n",
          ordstat_ptr->s_C_FIRST);
   fprintf(debug_fp,"\ts_C_MIDDLE     = %s\n",
          ordstat_ptr->s_C_MIDDLE);
   fprintf(debug_fp,"\ts_C_LAST       = %s\n",
          ordstat_ptr->s_C_LAST);
   fprintf(debug_fp,"\ts_C_BALANCE    = %.2f\n",
          ordstat_ptr->s_C_BALANCE);
   fprintf(debug_fp,"\ts_O_ID         = %d (%X)\n",
          ordstat_ptr->s_O_ID, ordstat_ptr->s_O_ID);
   fprintf(debug_fp,"\ts_O_ENTRY_D    = %s \n",
          ordstat_ptr->s_O_ENTRY_D_time);
   fprintf(debug_fp,"\ts_O_CARRIER_ID = %d (%X)\n",
          ordstat_ptr->s_O_CARRIER_ID, ordstat_ptr->s_O_CARRIER_ID);
   fprintf(debug_fp,"\ts_ol_cnt       = %d (%X)\n",
          ordstat_ptr->s_ol_cnt, ordstat_ptr->s_ol_cnt);
   fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
          ordstat_ptr->s_transtatus, ordstat_ptr->s_transtatus);
   fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
          ordstat_ptr->deadlocks, ordstat_ptr->deadlocks);

   fprintf(debug_fp,"\titems {\n");
   items = ordstat_ptr->s_ol_cnt;
   for (j = 0; j < items; j++) {
     if(j != 0)
       fprintf(debug_fp,"\n");
     fprintf(debug_fp,"\t\ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
          j, ordstat_ptr->item[j].s_OL_SUPPLY_W_ID, ordstat_ptr-
>item[j].s_OL_SUPPLY_W_ID);
     fprintf(debug_fp,"\t\ts_OL_I_ID[%d]        = %d (%X)\n",
          j, ordstat_ptr->item[j].s_OL_I_ID, ordstat_ptr->item[j].s_OL_I_ID);
     fprintf(debug_fp,"\t\ts_OL_QUANTITY[%d]    = %d (%X)\n",
          j, ordstat_ptr->item[j].s_OL_QUANTITY, ordstat_ptr->item[j].s_OL_QUANTITY);
     fprintf(debug_fp,"\t\ts_OL_AMOUNT[%d]      = %.2f\n",
          j, ordstat_ptr->item[j].s_OL_AMOUNT);
     fprintf(debug_fp,"\t\ts_OL_DELIVERY_D[%d]  = %s \n",
          j,ordstat_ptr->item[j].s_OL_DELIVERY_D_time);
   }
   fprintf(debug_fp,"\t}\n}\n\n");
   fclose(debug_fp);
}


/*---------------------------------------------------------------------*/
/* pay_debug                                                           */
/*---------------------------------------------------------------------*/
void pay_debug (struct out_payment_struct *payment_ptr,
                struct in_payment_struct  *in_payment,
                char *msg)
{
   char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

   InitializeDebug();
   strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
   strcat(debug_fn, "pay.debug.out");
   pay_print(payment_ptr, in_payment, debug_fn, msg);
}

/*---------------------------------------------------------------------*/
/* pay_print                                                           */
/*---------------------------------------------------------------------*/
void pay_print (struct out_payment_struct *payment_ptr,
                struct in_payment_struct  *in_payment,
                char *filename,
                char *msg)
{
   FILE *debug_fp;
   char  timeStamp[27];

   current_tmstmp(&timeStamp[0]);
   timeStamp[19] = (char)NULL;

   if ((debug_fp = fopen(filename, "a+")) == NULL)
   {
     return;
   }

   fprintf(debug_fp,"Payment debug information follows %s (%s)\n", timeStamp, msg);
   fprintf(debug_fp, " PID %d ", getpid());
   fprintf(debug_fp,"\n=================================================\n");

   fprintf(debug_fp,"in_payment_struct {\n");
   fprintf(debug_fp,"\ts_H_AMOUNT     = %.2f \n",
          in_payment->s_H_AMOUNT);
   fprintf(debug_fp,"\ts_C_ID         = %d (%X)\n",
          in_payment->s_C_ID, in_payment->s_C_ID);
   fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
          in_payment->s_W_ID, in_payment->s_W_ID);
   fprintf(debug_fp,"\ts_D_ID         = %d (%X)\n",
          in_payment->s_D_ID, in_payment->s_D_ID);
   fprintf(debug_fp,"\ts_C_D_ID       = %d (%X)\n",
          in_payment->s_C_D_ID, in_payment->s_C_D_ID);
   fprintf(debug_fp,"\ts_C_W_ID       = %d (%X)\n",
          in_payment->s_C_W_ID, in_payment->s_C_W_ID);
   fprintf(debug_fp,"\ts_C_LAST       = %s\n",
          in_payment->s_C_LAST);
   fprintf(debug_fp,"\n}\n\n");

   fprintf(debug_fp,"out_payment_struct {\n");
   fprintf(debug_fp,"\ts_C_CREDIT_LIM = %.2f\n",
          payment_ptr->s_C_CREDIT_LIM);
   fprintf(debug_fp,"\ts_C_DISCOUNT   = %04.4f\n",
          payment_ptr->s_C_DISCOUNT);
   fprintf(debug_fp,"\ts_C_BALANCE    = %.2f\n",
          payment_ptr->s_C_BALANCE);
   fprintf(debug_fp,"\ts_C_ID         = %d (%X)\n",
          payment_ptr->s_C_ID, payment_ptr->s_C_ID);
   fprintf(debug_fp,"\ts_W_STREET_1   = %s\n",
          payment_ptr->s_W_STREET_1);
   fprintf(debug_fp,"\ts_W_STREET_2   = %s\n",
          payment_ptr->s_W_STREET_2);
   fprintf(debug_fp,"\ts_W_CITY       = %s\n",
          payment_ptr->s_W_CITY);
   fprintf(debug_fp,"\ts_W_STATE      = %s\n",
          payment_ptr->s_W_STATE);
   fprintf(debug_fp,"\ts_W_ZIP        = %s\n",
          payment_ptr->s_W_ZIP);
```

```
  fprintf(debug_fp,"\ts_D_STREET_1   = %s\n",
        payment_ptr->s_D_STREET_1);
  fprintf(debug_fp,"\ts_D_STREET_2   = %s\n",
        payment_ptr->s_D_STREET_2);
  fprintf(debug_fp,"\ts_D_CITY       = %s\n",
        payment_ptr->s_D_CITY);
  fprintf(debug_fp,"\ts_D_STATE      = %s\n",
        payment_ptr->s_D_STATE);
  fprintf(debug_fp,"\ts_D_ZIP        = %s\n",
        payment_ptr->s_D_ZIP);
  fprintf(debug_fp,"\ts_C_FIRST      = %s\n",
        payment_ptr->s_C_FIRST);
  fprintf(debug_fp,"\ts_C_MIDDLE     = %s\n",
        payment_ptr->s_C_MIDDLE);
  fprintf(debug_fp,"\ts_C_LAST       = %s\n",
        payment_ptr->s_C_LAST);
  fprintf(debug_fp,"\ts_C_STREET_1   = %s\n",
        payment_ptr->s_C_STREET_1);
  fprintf(debug_fp,"\ts_C_STREET_2   = %s\n",
        payment_ptr->s_C_STREET_2);
  fprintf(debug_fp,"\ts_C_CITY       = %s\n",
        payment_ptr->s_C_CITY);
  fprintf(debug_fp,"\ts_C_STATE      = %s\n",
        payment_ptr->s_C_STATE);
  fprintf(debug_fp,"\ts_C_ZIP        = %s\n",
        payment_ptr->s_C_ZIP);
  fprintf(debug_fp,"\ts_C_PHONE      = %s\n",
        payment_ptr->s_C_PHONE);
  fprintf(debug_fp,"\ts_C_SINCE      = %s \n",
        payment_ptr->s_C_SINCE_time);
  fprintf(debug_fp,"\ts_C_CREDIT     = %s\n",
        payment_ptr->s_C_CREDIT);
  fprintf(debug_fp,"\ts_C_DATA       = %s\n",
        payment_ptr->s_C_DATA);
  fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
        payment_ptr->s_transtatus,payment_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
        payment_ptr->deadlocks,payment_ptr->deadlocks);
  fprintf(debug_fp,"\n}\n\n");
  fclose(debug_fp);
}


/*-------------------------------------------------------------*/
/* stk_debug                                                   */
/*-------------------------------------------------------------*/
void stk_debug (struct out_stocklev_struct *stocklev,
                struct in_stocklev_struct  *in_stocklev,
                char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "stk.debug.out");
  stk_print(stocklev, in_stocklev, debug_fn, msg);
}

/*-------------------------------------------------------------*/
/* stk_print                                                   */
/*-------------------------------------------------------------*/
void stk_print (struct out_stocklev_struct *stocklev,
                struct in_stocklev_struct  *in_stocklev,
                char *filename,
                char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Stock level debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==================================================\n");

  fprintf(debug_fp,"in_stocklev_struct {\n");
  fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
        in_stocklev->s_W_ID, in_stocklev->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID         = %d (%X)\n",
        in_stocklev->s_D_ID, in_stocklev->s_D_ID);
  fprintf(debug_fp,"\ts_threshold    = %d (%X)\n",
        in_stocklev->s_threshold, in_stocklev->s_threshold);
  fprintf(debug_fp,"}\n\n");

  fprintf(debug_fp,"out_stocklev_struct {\n");
  fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
        stocklev->s_transtatus, stocklev->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
        stocklev->deadlocks, stocklev->deadlocks);
  fprintf(debug_fp,"\ts_low_stock    = %d (%X)\n",
        stocklev->s_low_stock, stocklev->s_low_stock);
  fprintf(debug_fp,"}\n\n");
  fclose(debug_fp);
}

void current_tmstmp(char *buf)
{
  time_t t = time(NULL);
  strncpy(buf,ctime(&t),19);
}
```

include/db2tpcc.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * db2tpcc.h - Macros and Miscellany
 *
 */

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

/* ************************************************************************** */
/* Transaction Return Codes (s_transtatus)                                    */
/* ************************************************************************** */

#define INVALID_ITEM           100
#define TRAN_OK                0
#define FATAL_SQLERROR         -1

/* ************************************************************************** */
/* Definition of Unused and Bad Items                                         */
/* ************************************************************************** */
/* Define unused item ID to be 0. This allows the SUT to determine the     */
/* number of items in the order as required by 2.4.1.3 and 2.4.2.2 since   */
/* the assumption that any item with OL_I_ID = 0 is unused will be true.   */
/* This in turn requires that the value used for an invalid item is        */
/* equal to ITEMS + 1.                                                     */
/* ************************************************************************** */

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****************************************************************************/
/* NURand Constants                                                          */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.               */
/* Analysis indicates that a C_LAST delta of 85 is optimal.                  */
/*****************************************************************************/
#define C_C_LAST_RUN  88
#define C_C_LAST_LOAD 173
#define C_C_ID     319
#define C_OL_I_ID   3849
#define A_C_LAST    255
#define A_C_ID     1023
#define A_OL_I_ID   8191

/*****************************************************************************/
/* Transaction Type Identifiers                                              */
/*****************************************************************************/

#define CLIENT_SQL   0
#define NEWORD_SQL   1
#define PAYMENT_SQL  2
#define ORDSTAT_SQL  3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   in_items_struct {
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad1[3];
  } in_item[15];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  s_O_OL_CNT;    /* init by SUT */
  int16_t  s_all_local;
  int16_t  duplicate_items;
};

struct out_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   items_struct {
    float  s_I_PRICE;
    float  s_OL_AMOUNT;
    int16_t  s_S_QUANTITY;
    int16_t  pad2;
```

```
    char     s_I_NAME[25];
    char     s_brand_generic;
  } item[15];
  float  s_W_TAX;
  float  s_D_TAX;
  float  s_C_DISCOUNT;
  float  s_total_amount;
  int32_t  s_O_ID;
  int16_t  s_O_OL_CNT;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_LAST[17];
  char     s_C_CREDIT[3];
  char     s_O_ENTRY_D_time[27];

};

struct in_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  float  s_H_AMOUNT;
  int32_t  s_W_ID;
  int32_t  s_C_W_ID;
  int32_t  s_C_ID;
  int16_t  s_C_D_ID;
  int16_t  s_D_ID;
  char     s_C_LAST[17];
};

struct out_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  double  s_C_CREDIT_LIM;
  double  s_C_BALANCE;
  float  s_C_DISCOUNT;
  int32_t  s_C_ID;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_W_STREET_1[21];
  char     s_W_STREET_2[21];
  char     s_W_CITY[21];
  char     s_W_STATE[3];
  char     s_W_ZIP[10];
  char     s_D_STREET_1[21];
  char     s_D_STREET_2[21];
  char     s_D_CITY[21];
  char     s_D_STATE[3];
  char     s_D_ZIP[10];
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_C_STREET_1[21];
  char     s_C_STREET_2[21];
  char     s_C_CITY[21];
  char     s_C_STATE[3];
  char     s_C_ZIP[10];
  char     s_C_PHONE[17];
  char     s_C_CREDIT[3];
  char     s_C_DATA[201];
  char     s_H_DATE_time[27];
  char     s_C_SINCE_time[27];
};

struct in_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  pad1[3];
  char     s_C_LAST[17];
};

struct out_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  double  s_C_BALANCE;
  int32_t  s_C_ID;
  int32_t  s_O_ID;
  int16_t  s_O_CARRIER_ID;
  int16_t  s_ol_cnt;
  int16_t  pad1[2];
  struct   oitems_struct {
    double  s_OL_AMOUNT;
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad2;
    char     s_OL_DELIVERY_D_time[27];
  } item[15];
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_O_ENTRY_D_time[27];
  int16_t  pad3[2];
};

struct in_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_W_ID;
  int16_t  s_O_CARRIER_ID;
};

struct out_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_O_ID[10];
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

struct in_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_threshold;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
};

struct out_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_low_stock;
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

/* ******************************************************************** */
/* Transaction Prototypes                                               */
/* ******************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/* ******************************************************************** */
/* DB2 Connect/Disconnect & Thread Context Wrappers                     */
/* ******************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);


#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H
```

include/tpccapp.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpccapp.h - Application Macros
 */

#ifndef __TPCCAPP_H
#define __TPCCAPP_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include "sqlenv.h"
#define daricall __stdcall

#include "sqlca.h"
#include "sqlcodes.h"

#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var)  SwapEndian((void*)&Var, sizeof(Var))

/*****************************************************************************
  FUNCTION: SwapEndian
  PURPOSE: Swap the byte order of a structure
  EXAMPLE: int I=0x12345678; SWAP_BYTE(I); I => 0x78563412;
  IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op
          e.g.: *a = 0x12 [ Addr + 0];
```

```
            *b = 0x78 [ Add + 4 - 0 - 1 = Addr+3];
            *a ^= *b;         // sets *a to 0x6A
            *b ^= *a;         // sets *b to 0x12
            *a ^= *b;         // sets *a to 0x78

          Now *a => 0x78 && *b => 0x12
*****************************************************************************/

void SwapEndian(void *Addr, int nb)
{
  int i;
  for (i=0; i<nb/2; i++)
  {
    char *a = (char*)Addr+i;
    char *b = (char*)Addr+(nb-i-1);

    *a ^= *b;
    *b ^= *a;
    *a ^= *b;
  }
}
#endif   //SWAP_ENDIAN


/*****************************************************************************/
/* SQLCODE Macros                                                            */
/*****************************************************************************/

#define DLCHK(a) \
  if (sqlca.sqlcode == SQL_RC_E911) { goto a; }

/* ************************************************************************** */
/* In NOT ATOMIC COMPOUND SQL, all statements will be executed, but not    */
/* all will neccessarily complete successfully.  We can use sqlerrd(4) to  */
/* determine how many statements succeeded, but this won't tell us what    */
/* statements failed.  In order to determine this, we need to look at      */
/* sqlerrmc, which has the following structure:  HHHXNNNSSSSSXNNNSSSSS...  */
/* (See the docs for more details.)  Since we're interested in the first   */
/* failing statement, we can look at elements 5 and 6, which will contain  */
/* the first two digits of NNN (which is right-padded with spaces).  We    */
/* need to look at the first two digits since some of our compound blocks  */
/* have > 9 statements.  We convert these digits from ASCII to an int and  */
/* set 'last' to this value.                                               */
/* ************************************************************************** */

#define NACOMPCHK(last) \
  if (sqlca.sqlcode != SQL_RC_E1339) { last = -1; } \
  else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : sqlca.sqlerrmc[4]-0x30); \
         int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]-0x30); \
         if (b == 0) { last = a; } else { last = a * 10 + b; } \
       }

#endif // __TPCCAPP_H
```

include/lval.h

```
/* lval.h - generated automatically at 20060905.1052 */

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 41600
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H
```

include/tpccdbg.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpccdbg.h - Debugging Macros
 *
 */

#ifndef __TPCCDBG_H
#define __TPCCDBG_H

#ifdef __cplusplus
extern "C" {
#endif

extern void sqlerror (int tranType, char *msg, char *file, int line,
                      SQL_STRUCTURE sqlca *psqlca);

extern void new_debug (struct out_neword_struct *neword_ptr,
                       struct in_neword_struct *in_neword_ptr,
                       char *msg);
extern void pay_debug (struct out_payment_struct *payment_ptr,
                       struct in_payment_struct *in_payment_ptr,
                       char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                       struct in_ordstat_struct *in_ordstat_ptr,
                       char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
                       struct in_delivery_struct *in_delivery_ptr,
                       char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
                       struct in_stocklev_struct *in_stocklev_ptr,
                       char *msg);

extern void new_print (struct out_neword_struct *neword_ptr,
                       struct in_neword_struct *in_neword_ptr,
                       char *filename,
                       char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
                       struct in_payment_struct *in_payment_ptr,
                       char *filename,
                       char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
                       struct in_ordstat_struct *in_ordstat_ptr,
                       char *filename,
                       char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
                       struct in_delivery_struct *in_delivery_ptr,
                       char *filename,
                       char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
                       struct in_stocklev_struct *in_stocklev_ptr,
                       char *filename,
                       char *msg);

#ifdef __cplusplus
}
#endif

#endif // __TPCCDBG_H
```

tpccenv.bat

```
@REM ****************************************************************************
@REM  Licensed Materials - Property of IBM
@REM
@REM  Governed under the terms of the International
@REM  License Agreement for Non-Warranted Sample Code.
@REM
@REM  (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
@REM  All Rights Reserved.
@REM
@REM  US Government Users Restricted Rights - Use, duplication or
@REM  disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
@REM ****************************************************************************
@REM
@REM tpccenv.bat - Windows Environment Setup
@REM

@REM The Kit Version
set TPCC_VERSION=CK060815

@REM The DB2 Instance Name (for DB2)
set DB2INSTANCE=%USERNAME%

@REM The OS being used (i.e. "WINDOWS")
set PLATFORM=WINDOWS

@REM The type of make command and slash used by the OS
@REM (i.e. UNIX - "/", WINDOWS - "\")
@REM These are referenced all over the kit.
set SLASH=\
set MAKE=nmake

@REM Specifies whether or not to use dari stored proc's for the TPC-C driver.  Set to
either DARIVERSION or NONDARI;
@REM set TPCC_SPTYPE=NOSP
@REM set TPCC_SPTYPE=SPGENERAL2
set TPCC_SPTYPE=SPGENERAL
@REM set TPCC_SPTYPE=DARI2SQLDA

set DB2VERSION=v8

@REM The schema name is typically the SQL autorization ID (or username).
@REM This is required for runstats and EEE.
set TPCC_SCHEMA=%USERNAME%

@REM DB2 EE/EEE Configuration
set DB2EDITION=EE
@REM set DB2EDITION=EEE
set DB2NODE=0
@REM set to the number of nodes you have.  Set to 1 for EE.
set DB2NODES=1

@REM TPCC General Configuration
@REM ** IMPORTANT NOTE **
@REM The kit is not guaranteed to work properly if TPCC_ROOT or TPCC_SQLLIB
@REM have spaces in them.  If you absolutely must use paths with spaces,
@REM then the entire path must be surrounded by double quotes.
@REM For example:  HOME="C:\Program Files\IBM"
set HOME=C:\home\tpcc
set TPCC_DBNAME=TPCC
set TPCC_ROOT=%HOME%\tpc-c.ibm
set TPCC_SQLLIB=C:\Progra~1\IBM\sqllib
set TPCC_RUNDATA=%HOME%\tpc-c.ibm\tpccdata

@REM TPCC Debug Configuration
```

```
@REM This is the path where all error and debug logs are placed.
@REM To get debugging from within the stored procedures, you must
@REM set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
set TPCC_DEBUGDIR=c:\temp

@REM Specifies where stored procedures should be placed and if they should
@REM be fenced.
set TPCC_SPDIR=%TPCC_SQLLIB%\function
set TPCC_FENCED=NO
```

9.2. Transaction Code

Makefile.config

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
######

#
# Makefile.config - Linux 64-bit
#
#


# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g", "-DDEBUGIT" "-g -DDEBUGIT" or left blank
CC=cc
CFLAGS_OS=-DSQLUNIX -DSQLLinux -O2 -fpic -m64
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=gcc
LD_STORP=gcc
LDFLAGS_EXEC=
LDFLAGS_SHLIB=-shared
LDFLAGS_STORP=$(LDFLAGS_SHLIB)
LDFLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2 -m64
LDFLAGS_OUT=-o

# Library Configuration
AR=ar
ARFLAGS=-rv
ARFLAGS_LIB=
ARFLAGS_OUT=

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

# OS File Extensions & Path Separators
OBJEXT=.o
LIBEXT=.a
SHLIBEXT=.so
BINEXT=
SLASH=/
CMDSEP=;
```

Src.Common/Makefile

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
######

#
# Makefile - Makefile for Src.Common
#
#

include $(TPCC_ROOT)/Makefile.config

#
################################################################################
# Preprocessor, Compiler and LInker Flags
#
################################################################################

BND_OPTS =    GRANT PUBLIC \
        MESSAGES $*.bnd.msg
PRP_OPTS =    BINDFILE \
        OPTLEVEL 1 \
        ISOLATION RR \
        MESSAGES $*.prep.msg \
        LEVEL $(TPCC_VERSION) \
        NOLINEMACRO

INCLUDE =   -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =    $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDE) \
        -DSQLA_NOLINES -D$(DB2EDITION) -D$(DB2VERSION) \
        -D$(TPCC_SPTYPE)

UTIL_OBJ_DBG =tpccdbg$(OBJEXT)
UTIL_OBJ_GEN =tpccmisc$(OBJEXT)
UTIL_OBJ_DB2 = tpccctx$(OBJEXT)

#
################################################################################
# User Targets
#
################################################################################

all:    $(UTIL_OBJ_DBG) $(UTIL_OBJ_GEN) connect $(UTIL_OBJ_DB2) disconnect

dbgen:  $(UTIL_OBJ_GEN)

clean:
    - $(ERASE) *$(OBJEXT) *.bnd *.msg tpccctx.c

#
################################################################################
# Helper Targets
#
################################################################################

connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

rebind:  connect
    db2 bind tpccctx.bnd $(BND_OPTS)

#
################################################################################
# Build Rules
#
################################################################################

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
    @echo "Prepping $*.sqc"
    -db2 prep $*.sqc $(PRP_OPTS)
    @echo "Binding $*.bnd"
    db2 bind $*.bnd $(BND_OPTS)

#
################################################################################
# Dependencies
#
################################################################################

# Source
tpccdbg$(OBJEXT):   tpccdbg.c
tpccctx$(OBJEXT):    tpccctx.c
tpccmisc$(OBJEXT): tpccmisc.c

# Headers
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Common/tpccctx.sqc

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 *
 * tpccctx.sqc - TPCC context code
 *
 */
```

```
#include <string.h>
#include <sqlutil.h>
#include "db2tpcc.h"
#include "tpccdbg.h"

int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username, char *in_password);
int disconnect_from_TM(void);

int connect_to_TM(char *in_dbname)
{
  return connect_to_TM_auth(in_dbname, "", "");
}

int connect_to_TM_auth(char *in_dbname, char *in_username, char *in_password)
{
   SQL_STRUCTURE sqlca  sqlca;
   int ConnectSQLCODE = 0;

   EXEC SQL BEGIN DECLARE SECTION;
   char dbname[9];
   char username[129];
   char password[15];
   EXEC SQL END DECLARE SECTION;

   /* Copy 9 characters - 8 for dbname, 1 for NULL */
   strncpy(dbname,in_dbname,9);
   if (strcmp(in_username,"") == 0)
   {
      EXEC SQL CONNECT TO :dbname IN SHARE MODE;
   } else {
      strncpy(username,in_username,128);
      strncpy(password,in_password,14);
      EXEC SQL CONNECT TO :dbname IN SHARE MODE USER :username
USING :password;
   }

   ConnectSQLCODE = SQLCODE;
   if (ConnectSQLCODE != 0)
   {
      sqlerror( CLIENT_SQL, "CONNECT", __FILE__, __LINE__, &sqlca);

      return ConnectSQLCODE;
   }

   return 0;
}

int disconnect_from_TM(void)
{
   SQL_STRUCTURE sqlca  sqlca;
   int DisconnectSQLCODE = 0;

   EXEC SQL CONNECT RESET;

   DisconnectSQLCODE = SQLCODE;
   if (DisconnectSQLCODE != 0) {
      sqlerror( CLIENT_SQL, "DISCONNECT", __FILE__, __LINE__, &sqlca);
   }

   if (DisconnectSQLCODE) {
      return DisconnectSQLCODE;
   }
   return 0;
}
```

Src.Common/tpccdbg.c

```
/*********************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*********************************************************************************/

/*
 * tccdbg.c - Debugging Routines
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>

#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"

#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128

void  del_print();
void  new_print();
void  ord_print();
void  pay_print();
void  stk_print();

void current_tmstmp(char *buf);

static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";

/*----------------------------------------------------------------*/
/*       InitializeDebug                                          */
/*----------------------------------------------------------------*/
__inline void InitializeDebug(void) {
   if (debugInit == 0) {
      char *p = getenv("TPCC_DEBUGDIR");
      if (p) {
         strncpy(debugPath, p, DEBUG_PATH_SIZE);
      } else {
         strcpy(debugPath, "/tmp");
      }
      strcat(debugPath, "/");
   }
   debugInit = 1;
}

/*----------------------------------------------------------------*/
/*       sqlerror                                                 */
/*----------------------------------------------------------------*/
void sqlerror(int tranType, char *msg, char *file, int line, SQL_STRUCTURE sqlca *psqlca)
{
   FILE *err_fp = NULL;
   char  err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
   char  tranName[16];
   int j,k;
   char  timeStamp[27];
   char  errStr[512] = "";

   InitializeDebug();
   strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
   current_tmstmp(&timeStamp[0]);
   timeStamp[19] = (char)NULL;

   switch(tranType)
   {
     case NEWORD_SQL:
        // sprintf(err_fn, "%d.err.out", getpid());
        strcat(err_fn, "new.err.out");
        strcpy(tranName, "NEW_ORDER");
        break;

     case DELIVERY_SQL:
        // sprintf(err_fn, "%d.err.out", getpid());
        strcat(err_fn, "del.err.out");
        strcpy(tranName, "DELIVERY");
        break;

     case PAYMENT_SQL:
        // sprintf(err_fn, "%d.err.out", getpid());
        strcat(err_fn, "pay.err.out");
        strcpy(tranName, "PAYMENT");
        break;

     case ORDSTAT_SQL:
        // sprintf(err_fn, "%d.err.out", getpid());
        strcat(err_fn, "ord.err.out");
        strcpy(tranName, "ORDER_STAT");
        break;

     case STOCKLEV_SQL:
        //sprintf(err_fn, "%d.err.out", getpid());
        strcat(err_fn, "stk.err.out");
        strcpy(tranName, "STOCK_LVL");
        break;

     case 0:
        strcat(err_fn, "cli.err.out");
        strcpy(tranName, "CLIENT");
        break;

     default:
        return;

   }

   /* Generate Formatted Error Message */
   sqlaintp(errStr, 512, 78, psqlca);

   if ((err_fp = fopen(err_fn, "a+")) == NULL)
   {
     return;
   }

   fprintf(err_fp, "----------------------------------------\n");
   fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
   fprintf(err_fp, "FILE %s (%u)\n", file, line);
   fprintf(err_fp, "SQLCODE %d ", psqlca->sqlcode);
   fprintf(err_fp, "PID %d ", getpid());
   fprintf(err_fp, "TIME %s\n", timeStamp);
   fprintf(err_fp, "----------------------------------------\n");
   fprintf(err_fp, "%s", errStr);
   fprintf(err_fp, "----------------------------------------\n");

   if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
   {
     fprintf(err_fp, "slerrmc:  ");
```

```
    for(j = 0; j < 5; j++)
    {
      for(k = 0; k < 16; k++) {
        int pos = j * 16 + k;
        if (pos < 70) fprintf(err_fp, "%02x ", psqlca->sqlerrmc[pos]);
        else fprintf(err_fp, "   ");
      }
      fprintf(err_fp,"   |");
      for(k = 0; k < 16; k++) {
        int pos = j * 16 + k;
        char c = ' ';
        if (pos < 70) {
          c = psqlca->sqlerrmc[pos];
          if (!isprint(c)) c = ' ';
        }
        fprintf(err_fp,"%c", c);
      }
      fprintf(err_fp,"|\n");
      if (j < 4) fprintf(err_fp,"          ");
    }
  }

  fprintf(err_fp, "sqlerrp:  ");
  for(j = 0; j < 8; j++)
    fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
  fprintf(err_fp,"\n");

  fprintf(err_fp, "sqlerrd: ");
  for(j = 0; j < 6; j++)
    fprintf(err_fp, " %d", psqlca->sqlerrd[j]);
  fprintf(err_fp,"\n");

  if (psqlca->sqlwarn[0] != ' ')
  {
    fprintf(err_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
      fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
    fprintf(err_fp,"\n");
  }

  fprintf(err_fp, "\n");

  fclose(err_fp);
}


/*------------------------------------------------------------------*/
/* del_debug                                                        */
/*------------------------------------------------------------------*/
void del_debug (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct  *in_delivery,
                char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "del.debug.out");
  del_print(delivery_ptr, in_delivery, debug_fn, msg);
}

/*------------------------------------------------------------------*/
/* del_print                                                        */
/*------------------------------------------------------------------*/
void del_print (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct  *in_delivery,
                char *filename,
                char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];
  int   j;

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Delivery debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n================================================\n");

  fprintf(debug_fp,"in_delivery_struct {\n");
  fprintf(debug_fp,"\ts_W_ID        = %d (%X)\n",
          in_delivery->s_W_ID, in_delivery->s_W_ID);
  fprintf(debug_fp,"\ts_O_CARRIER_ID = %d (%X)\n",
          in_delivery->s_O_CARRIER_ID, in_delivery->s_O_CARRIER_ID);
  fprintf(debug_fp,"}\n\n");

  fprintf(debug_fp,"out_delivery_struct {\n");
  fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
          delivery_ptr->s_transtatus,delivery_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
          delivery_ptr->deadlocks,delivery_ptr->deadlocks);

  for (j = 0; j < 10; j++) {
    fprintf(debug_fp,"\t\ts_O_ID[%d]         = %d\n",
            j, delivery_ptr->s_O_ID[j]);
  }
  fprintf(debug_fp,"\t}\n}\n\n");
  fclose(debug_fp);
}


/*------------------------------------------------------------------*/
/* new_debug                                                        */
/*------------------------------------------------------------------*/
void new_debug (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "new.debug.out");
  new_print(neword_ptr, in_neword, debug_fn, msg);
}

/*------------------------------------------------------------------*/
/* new_print                                                        */
/*------------------------------------------------------------------*/
void new_print (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *filename,
                char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];
  int   j, items;

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"New order debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n================================================\n");

  fprintf(debug_fp,"in_neword_struct {\n");

  fprintf(debug_fp,"\ts_C_ID         = %d (%X)\n",
          in_neword->s_C_ID, in_neword->s_C_ID);
  fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
          in_neword->s_W_ID, in_neword->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID         = %d (%X)\n",
          in_neword->s_D_ID, in_neword->s_D_ID);
  fprintf(debug_fp,"\ts_O_OL_CNT     = %d (%X)\n",
          in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
  fprintf(debug_fp,"\ts_all_local    = %d (%X)\n",
          in_neword->s_all_local, in_neword->s_all_local);
// fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
//         in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp,"\tduplicate_items= %d (%X)\n",
//         in_neword->duplicate_items, in_neword->duplicate_items);

  fprintf(debug_fp,"\titems {\n");
  items = in_neword->s_O_OL_CNT;
  for (j=0; j<items; j++) {
    if(j != 0)
      fprintf(debug_fp,"\n");
    fprintf(debug_fp,"\t\ts_OL_I_ID[%d]        = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_I_ID, in_neword->in_item[j].s_OL_I_ID);
    fprintf(debug_fp,"\t\ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_SUPPLY_W_ID, in_neword-
>in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp,"\t\ts_OL_QUANTITY[%d]    = %d (%X)\n",
            j, in_neword->in_item[j].s_OL_QUANTITY, in_neword-
>in_item[j].s_OL_QUANTITY);
  }
  fprintf(debug_fp,"\t}\n}\n\n");

  fprintf(debug_fp,"out_neword_struct {\n");
  fprintf(debug_fp,"\ts_C_LAST       = %s\n",
          neword_ptr->s_C_LAST);
  fprintf(debug_fp,"\ts_C_CREDIT     = %s\n",
          neword_ptr->s_C_CREDIT);
  fprintf(debug_fp,"\ts_W_TAX        = %04.4f \n",
          neword_ptr->s_W_TAX);
  fprintf(debug_fp,"\ts_D_TAX        = %04.4f \n",
          neword_ptr->s_D_TAX);
  fprintf(debug_fp,"\ts_C_DISCOUNT   = %04.4f \n",
          neword_ptr->s_C_DISCOUNT);
  fprintf(debug_fp,"\ts_O_ID         = %d (%X)\n",
          neword_ptr->s_O_ID, neword_ptr->s_O_ID);
  fprintf(debug_fp,"\ts_O_OL_CNT     = %d (%X)\n",
          neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
  fprintf(debug_fp,"\ts_O_ENTRY_D    = %s \n",
          neword_ptr->s_O_ENTRY_D_time);
  fprintf(debug_fp,"\ts_total_amount = %.2f \n",
          neword_ptr->s_total_amount);
  fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
          neword_ptr->s_transtatus, neword_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
          neword_ptr->deadlocks, neword_ptr->deadlocks);

// fprintf(debug_fp,"\ts_W_ID         = %d (%X)\n",
//         neword_ptr->s_W_ID, neword_ptr->s_W_ID);
// fprintf(debug_fp,"\ts_D_ID         = %d (%X)\n",
//         neword_ptr->s_D_ID, neword_ptr->s_D_ID);
// fprintf(debug_fp,"\ts_all_local    = %d (%X)\n",
//         neword_ptr->s_all_local, neword_ptr->s_all_local);
```

```
// fprintf(debug_fp,"\tduplicate_items= %d (%X)\n",
//         neword_ptr->duplicate_items, neword_ptr->duplicate_items);

   fprintf(debug_fp,"\titems {\n");
   items = neword_ptr->s_O_OL_CNT;
   for (j=0; j<items; j++) {
     if(j != 0)
       fprintf(debug_fp,"\n");
     fprintf(debug_fp,"\t\ts_I_NAME[%d]         = %s\n",
             j, neword_ptr->item[j].s_I_NAME);
     fprintf(debug_fp,"\t\ts_I_PRICE[%d]        = %.2f \n",
             j, neword_ptr->item[j].s_I_PRICE);
     fprintf(debug_fp,"\t\ts_OL_AMOUNT[%d]      = %.2f \n",
             j, neword_ptr->item[j].s_OL_AMOUNT);
     fprintf(debug_fp,"\t\ts_S_QUANTITY[%d]     = %d (%X)\n",
             j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr->item[j].s_S_QUANTITY);
     fprintf(debug_fp,"\t\ts_brand_generic[%d]  = %c\n",
             j, neword_ptr->item[j].s_brand_generic);
   }
   fprintf(debug_fp,"\t}\n}\n\n");
   fclose(debug_fp);
}



/*---------------------------------------------------------------*/
/* ord_debug                                                     */
/*---------------------------------------------------------------*/
void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                struct in_ordstat_struct  *in_ordstat,
                char *msg)
{
   char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

   InitializeDebug();
   strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
   strcat(debug_fn, "ord.debug.out");
   ord_print(ordstat_ptr, in_ordstat, debug_fn, msg);
}

/*---------------------------------------------------------------*/
/* ord_print                                                     */
/*---------------------------------------------------------------*/
void ord_print (struct out_ordstat_struct *ordstat_ptr,
                struct in_ordstat_struct  *in_ordstat,
                char *filename,
                char *msg)
{
   FILE *debug_fp;
   char  timeStamp[27];
   int   j, items;

   current_tmstmp(&timeStamp[0]);
   timeStamp[19] = (char)NULL;

   if ((debug_fp = fopen(filename, "a+")) == NULL)
   {
     return;
   }

   fprintf(debug_fp,"Order status debug information follows %s (%s)\n", timeStamp, msg);
   fprintf(debug_fp, " PID %d ", getpid());
   fprintf(debug_fp,"\n==================================================\n");

   fprintf(debug_fp,"in_ordstat_struct {\n");
   fprintf(debug_fp,"\ts_W_ID          = %d (%X)\n",
           in_ordstat->s_W_ID, in_ordstat->s_W_ID);
   fprintf(debug_fp,"\ts_D_ID          = %d (%X)\n",
           in_ordstat->s_D_ID, in_ordstat->s_D_ID);
   fprintf(debug_fp,"\ts_C_ID          = %d (%X)\n",
           in_ordstat->s_C_ID, in_ordstat->s_C_ID);
   fprintf(debug_fp,"\ts_C_LAST        = %s\n",
           in_ordstat->s_C_LAST);
   fprintf(debug_fp,"}\n\n");

   fprintf(debug_fp,"out_ordstat_struct {\n");
   fprintf(debug_fp,"\ts_C_ID          = %d (%X)\n",
           ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);
   fprintf(debug_fp,"\ts_C_FIRST       = %s\n",
           ordstat_ptr->s_C_FIRST);
   fprintf(debug_fp,"\ts_C_MIDDLE      = %s\n",
           ordstat_ptr->s_C_MIDDLE);
   fprintf(debug_fp,"\ts_C_LAST        = %s\n",
           ordstat_ptr->s_C_LAST);
   fprintf(debug_fp,"\ts_C_BALANCE     = %.2f\n",
           ordstat_ptr->s_C_BALANCE);
   fprintf(debug_fp,"\ts_O_ID          = %d (%X)\n",
           ordstat_ptr->s_O_ID, ordstat_ptr->s_O_ID);
   fprintf(debug_fp,"\ts_O_ENTRY_D     = %s \n",
           ordstat_ptr->s_O_ENTRY_D_time);
   fprintf(debug_fp,"\ts_O_CARRIER_ID = %d (%X)\n",
           ordstat_ptr->s_O_CARRIER_ID, ordstat_ptr->s_O_CARRIER_ID);
   fprintf(debug_fp,"\ts_ol_cnt        = %d (%X)\n",
           ordstat_ptr->s_ol_cnt, ordstat_ptr->s_ol_cnt);
   fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
           ordstat_ptr->s_transtatus, ordstat_ptr->s_transtatus);
   fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
           ordstat_ptr->deadlocks, ordstat_ptr->deadlocks);

   fprintf(debug_fp,"\titems {\n");
   items = ordstat_ptr->s_ol_cnt;
   for (j = 0; j < items; j++) {
     if(j != 0)
       fprintf(debug_fp,"\n");
     fprintf(debug_fp,"\t\ts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
             j, ordstat_ptr->item[j].s_OL_SUPPLY_W_ID, ordstat_ptr-
>item[j].s_OL_SUPPLY_W_ID);
     fprintf(debug_fp,"\t\ts_OL_I_ID[%d]        = %d (%X)\n",
             j, ordstat_ptr->item[j].s_OL_I_ID, ordstat_ptr->item[j].s_OL_I_ID);
     fprintf(debug_fp,"\t\ts_OL_QUANTITY[%d]    = %d (%X)\n",
             j, ordstat_ptr->item[j].s_OL_QUANTITY, ordstat_ptr->item[j].s_OL_QUANTITY);
     fprintf(debug_fp,"\t\ts_OL_AMOUNT[%d]      = %.2f\n",
             j, ordstat_ptr->item[j].s_OL_AMOUNT);
     fprintf(debug_fp,"\t\ts_OL_DELIVERY_D[%d]  = %s \n",
             j,ordstat_ptr->item[j].s_OL_DELIVERY_D_time);
   }
   fprintf(debug_fp,"\t}\n}\n\n");
   fclose(debug_fp);
}


/*---------------------------------------------------------------*/
/* pay_debug                                                     */
/*---------------------------------------------------------------*/
void pay_debug (struct out_payment_struct *payment_ptr,
                struct in_payment_struct  *in_payment,
                char *msg)
{
   char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

   InitializeDebug();
   strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
   strcat(debug_fn, "pay.debug.out");
   pay_print(payment_ptr, in_payment, debug_fn, msg);
}

/*---------------------------------------------------------------*/
/* pay_print                                                     */
/*---------------------------------------------------------------*/
void pay_print (struct out_payment_struct *payment_ptr,
                struct in_payment_struct  *in_payment,
                char *filename,
                char *msg)
{
   FILE *debug_fp;
   char  timeStamp[27];

   current_tmstmp(&timeStamp[0]);
   timeStamp[19] = (char)NULL;

   if ((debug_fp = fopen(filename, "a+")) == NULL)
   {
     return;
   }

   fprintf(debug_fp,"Payment debug information follows %s (%s)\n", timeStamp, msg);
   fprintf(debug_fp, " PID %d ", getpid());
   fprintf(debug_fp,"\n==================================================\n");

   fprintf(debug_fp,"in_payment_struct {\n");
   fprintf(debug_fp,"\ts_H_AMOUNT      = %.2f \n",
           in_payment->s_H_AMOUNT);
   fprintf(debug_fp,"\ts_C_ID          = %d (%X)\n",
           in_payment->s_C_ID, in_payment->s_C_ID);
   fprintf(debug_fp,"\ts_W_ID          = %d (%X)\n",
           in_payment->s_W_ID, in_payment->s_W_ID);
   fprintf(debug_fp,"\ts_D_ID          = %d (%X)\n",
           in_payment->s_D_ID, in_payment->s_D_ID);
   fprintf(debug_fp,"\ts_C_D_ID        = %d (%X)\n",
           in_payment->s_C_D_ID, in_payment->s_C_D_ID);
   fprintf(debug_fp,"\ts_C_W_ID        = %d (%X)\n",
           in_payment->s_C_W_ID, in_payment->s_C_W_ID);
   fprintf(debug_fp,"\ts_C_LAST        = %s\n",
           in_payment->s_C_LAST);
   fprintf(debug_fp,"\n}\n\n");

   fprintf(debug_fp,"out_payment_struct {\n");
   fprintf(debug_fp,"\ts_C_CREDIT_LIM = %.2f\n",
           payment_ptr->s_C_CREDIT_LIM);
   fprintf(debug_fp,"\ts_C_DISCOUNT   = %04.4f\n",
           payment_ptr->s_C_DISCOUNT);
   fprintf(debug_fp,"\ts_C_BALANCE    = %.2f\n",
           payment_ptr->s_C_BALANCE);
   fprintf(debug_fp,"\ts_C_ID          = %d (%X)\n",
           payment_ptr->s_C_ID, payment_ptr->s_C_ID);
   fprintf(debug_fp,"\ts_W_STREET_1   = %s\n",
           payment_ptr->s_W_STREET_1);
   fprintf(debug_fp,"\ts_W_STREET_2   = %s\n",
           payment_ptr->s_W_STREET_2);
   fprintf(debug_fp,"\ts_W_CITY       = %s\n",
           payment_ptr->s_W_CITY);
   fprintf(debug_fp,"\ts_W_STATE      = %s\n",
           payment_ptr->s_W_STATE);
   fprintf(debug_fp,"\ts_W_ZIP        = %s\n",
           payment_ptr->s_W_ZIP);
   fprintf(debug_fp,"\ts_D_STREET_1   = %s\n",
           payment_ptr->s_D_STREET_1);
   fprintf(debug_fp,"\ts_D_STREET_2   = %s\n",
           payment_ptr->s_D_STREET_2);
   fprintf(debug_fp,"\ts_D_CITY       = %s\n",
           payment_ptr->s_D_CITY);
   fprintf(debug_fp,"\ts_D_STATE      = %s\n",
           payment_ptr->s_D_STATE);
   fprintf(debug_fp,"\ts_D_ZIP        = %s\n",
           payment_ptr->s_D_ZIP);
   fprintf(debug_fp,"\ts_C_FIRST      = %s\n",
           payment_ptr->s_C_FIRST);
   fprintf(debug_fp,"\ts_C_MIDDLE     = %s\n",
           payment_ptr->s_C_MIDDLE);
```

```
  fprintf(debug_fp,"\ts_C_LAST       = %s\n",
        payment_ptr->s_C_LAST);
  fprintf(debug_fp,"\ts_C_STREET_1   = %s\n",
        payment_ptr->s_C_STREET_1);
  fprintf(debug_fp,"\ts_C_STREET_2   = %s\n",
        payment_ptr->s_C_STREET_2);
  fprintf(debug_fp,"\ts_C_CITY       = %s\n",
        payment_ptr->s_C_CITY);
  fprintf(debug_fp,"\ts_C_STATE      = %s\n",
        payment_ptr->s_C_STATE);
  fprintf(debug_fp,"\ts_C_ZIP        = %s\n",
        payment_ptr->s_C_ZIP);
  fprintf(debug_fp,"\ts_C_PHONE      = %s\n",
        payment_ptr->s_C_PHONE);
  fprintf(debug_fp,"\ts_C_SINCE      = %s \n",
        payment_ptr->s_C_SINCE_time);
  fprintf(debug_fp,"\ts_C_CREDIT     = %s\n",
        payment_ptr->s_C_CREDIT);
  fprintf(debug_fp,"\ts_C_DATA       = %s\n",
        payment_ptr->s_C_DATA);
  fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
        payment_ptr->s_transtatus,payment_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
        payment_ptr->deadlocks,payment_ptr->deadlocks);
  fprintf(debug_fp,"\n}\n\n");
  fclose(debug_fp);
}


/*-------------------------------------------------------------*/
/* stk_debug                                                   */
/*-------------------------------------------------------------*/
void stk_debug (struct out_stocklev_struct *stocklev,
            struct in_stocklev_struct  *in_stocklev,
            char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "stk.debug.out");
  stk_print(stocklev, in_stocklev, debug_fn, msg);
}

/*-------------------------------------------------------------*/
/* stk_print                                                   */
/*-------------------------------------------------------------*/
void stk_print (struct out_stocklev_struct *stocklev,
            struct in_stocklev_struct  *in_stocklev,
            char *filename,
            char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Stock level debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n=================================================\n");

  fprintf(debug_fp,"in_stocklev_struct {\n");
  fprintf(debug_fp,"\ts_W_ID        = %d (%X)\n",
        in_stocklev->s_W_ID, in_stocklev->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID        = %d (%X)\n",
        in_stocklev->s_D_ID, in_stocklev->s_D_ID);
  fprintf(debug_fp,"\ts_threshold    = %d (%X)\n",
        in_stocklev->s_threshold, in_stocklev->s_threshold);
  fprintf(debug_fp,"}\n\n");

  fprintf(debug_fp,"out_stocklev_struct {\n");
  fprintf(debug_fp,"\ts_transtatus   = %d (%X)\n",
        stocklev->s_transtatus, stocklev->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      = %d (%X)\n",
        stocklev->deadlocks, stocklev->deadlocks);
  fprintf(debug_fp,"\ts_low_stock    = %d (%X)\n",
        stocklev->s_low_stock, stocklev->s_low_stock);
  fprintf(debug_fp,"}\n\n");
  fclose(debug_fp);
}

void current_tmstmp(char *buf)
{
  time_t t = time(NULL);
  strncpy(buf,ctime(&t),19);
}
```

Src.Common/tpccmisc.c

```
/*********************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*********************************************************************************/

/*
 *
 * tpccmisc.c - Miscellaneous routines
 *
 */

#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>

double current_time_ms(void);
double current_time(void);


/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
  /* use time() to get seconds */
  return(time(NULL));
}

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
   /* gettimeofday() returns seconds and microseconds */
   /* convert to fractional seconds */
  struct timeval t;
  gettimeofday(&t,NULL);
  return (t.tv_sec + (double)t.tv_usec/(1000*1000));
}
```

Src.Srv/Makefile

```
#################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#################################################################################
######

#
# Makefile - Makefile for Src.Srv
#
#

include $(TPCC_ROOT)/Makefile.config

#
#################################################################################
# Preprocessor, Compiler and Linker Flags
#
#################################################################################

BND_OPTS =    GRANT PUBLIC \
        MESSAGES $*.bnd.msg
PRP_OPTS =    BINDFILE \
        EXPLAIN ALL \
        MESSAGES $*.prep.msg

INCLUDE =       -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =    $(CFLAGS_OS) $(INCLUDE) $(CFLAGS_DEBUG) \
        -D$(DB2EDITION) -D$(DB2VERSION) \
        -DSQLA_NOLINES -DLINT_ARGS

LDFLAGS =  $(LDFLAGS_STORP) $(LDFLAGS_LIB)

#
#################################################################################
# File Collections
#
#################################################################################

STORED_PROCS =  new ord del

UTIL_OBJ =  $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
        $(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT)

EXE =       news ords dels

#
#################################################################################
# User Targets
#
#################################################################################

all:    connect explain catalog $(EXE) install plan disconnect

clean:    connect uncatalog unexplain disconnect
    - $(ERASE) $(TPCC_SPDIR)$(SLASH)news
    - $(ERASE) $(TPCC_SPDIR)$(SLASH)ords
    - $(ERASE) $(TPCC_SPDIR)$(SLASH)dels
    - $(ERASE) *.bnd *.msg *.out *$(OBJEXT) $(EXE) tpcc_all_sql.c
```

```
    - $(ERASE) TPCC_ALL.*.plan

#
################################################################################
# Helper Targets
#
################################################################################

catalog:uncatalog
    - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)genproc.pl $(STORED_PROCS)
    - db2 -tvf cat-proc.ddl +o -z cat-proc.out
    - db2 -td% -vf cat-func.ddl +o -z cat-func.out

uncatalog:
    - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)genproc.pl $(STORED_PROCS)
    - db2 -td% -vf uncat-func.ddl +o -z uncat-func.out
    - db2 -tvf uncat-proc.ddl +o -z uncat-proc.out

explain:
    - perl $(TPCC_ROOT)$(SLASH)utils$(SLASH)fixup_explain.pl
    - db2 -tvf $(TPCC_ROOT)$(SLASH)utils$(SLASH)EXPLAIN.DDL +o -z EXPLAIN.out

unexplain:
    - db2 -tvf $(TPCC_ROOT)$(SLASH)utils$(SLASH)UNEXPLAIN.DDL +o -z
UNEXPLAIN.out

connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

plan:
    - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s $(TPCC_SCHEMA) -w -1 -n
TPCC_ALL -g -# 0 -o TPCC_ALL.exfmt.plan
    - (export DB2EXPLN_BUFFER=3000000; db2expln -d $(TPCC_DBNAME) -c
$(TPCC_SCHEMA) -p TPCC_ALL -s 0 -g -o TPCC_ALL.expln.plan )

rebind:  connect catalog
    db2 bind tpcc_all_sql.bnd $(BND_OPTS) QUERYOPT 7

#
################################################################################
# Install Targets
#
################################################################################

install:  $(EXE)
    - mkdir $(TPCC_SPDIR)
    $(COPY) ords $(TPCC_SPDIR)
    $(COPY) news $(TPCC_SPDIR)
    $(COPY) dels $(TPCC_SPDIR)

#
################################################################################
# Build Rules
#
################################################################################

.SUFFIXES: $(OBJEXT) .c .sqc

# d230437mte: QUERYOPT 7 required for UNION ALL
# Only stock needs CS , and that can be specified on the SELECT statement
tpcc_all_sql.c:
    @echo "Prepping $*.sqc"
    -db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
    @echo "Binding $*.bnd"
    db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7

# Stored procedures are built in a special way

tpcc_all_sql$(OBJEXT):
    $(CC) -c tpcc_all_sql.c $(CFLAGS) -D$(TPCC_SPTYPE) $(CFLAGS_OUT)$@

$(EXE): $(UTIL_OBJ) tpcc_all_sql.o
    $(LD_STORP) $(LDFLAGS) $(UTIL_OBJ) tpcc_all_sql.o $(LDFLAGS_OUT)$@

#
################################################################################
# Dependencies
#
################################################################################

# Executables (Stored Procedures)
$(EXE): $(UTIL_OBJ) tpcc_all_sql.o

# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c

# Headers
tpcc_all_sql.c:   $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Srv/cat-func.ddl

```
-------------------------------------------------------------------------------
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-------------------------------------------------------------------------------
--
-- cat-func.ddl - Create table functions
--
--


--
-- DELIVERY
--

CREATE FUNCTION DEL(   W_ID          INTEGER
                     , D_ID          SMALLINT
                     , CARRIER_ID    SMALLINT
                   )

RETURNS TABLE ( O_ID INTEGER )

SPECIFIC DELIVERY

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE O_ID   INTEGER ;
  DECLARE C_ID   INTEGER ;
  DECLARE AMOUNT DECIMAL(12,2) ;

  /* Delete the order from new order table */

  SET VAR.O_ID = ( SELECT NO_O_ID

                     FROM OLD TABLE ( DELETE
                                        FROM ( SELECT NO_O_ID

                                                 FROM NEW_ORDER

                                                WHERE NO_W_ID = DEL.W_ID
                                                  AND NO_D_ID = DEL.D_ID

                                                ORDER BY NO_O_ID ASC

                                                FETCH FIRST 1 ROW ONLY
                                              ) AS NEW_ORDER
                                      ) AS D
                   )
  ;

  /* Update the order as delivered and retrieve the customer id */

  SET VAR.C_ID = ( SELECT O_C_ID

                     FROM OLD TABLE ( UPDATE ORDERS

                                         SET O_CARRIER_ID = DEL.CARRIER_ID

                                       WHERE O_W_ID = DEL.W_ID
                                         AND O_D_ID = DEL.D_ID
                                         AND O_ID   = VAR.O_ID
                                      ) AS U
                   )
  ;

  SET VAR.AMOUNT = ( SELECT SUM( OL_AMOUNT )

                       FROM OLD TABLE ( UPDATE ORDER_LINE

                                           SET OL_DELIVERY_D = CURRENT TIMESTAMP

                                         WHERE OL_W_ID = DEL.W_ID
                                           AND OL_D_ID = DEL.D_ID
                                           AND OL_O_ID = VAR.O_ID
                                      ) AS U
                     )
  ;

  /* Charge the customer */

  UPDATE CUSTOMER

     SET   C_BALANCE      = C_BALANCE + VAR.AMOUNT
         , C_DELIVERY_CNT = C_DELIVERY_CNT + SMALLINT( 1 )

   WHERE C_W_ID = DEL.W_ID
     AND C_D_ID = DEL.D_ID
     AND C_ID   = VAR.C_ID
  ;

  /* Return the order id to the caller (or NULL) */

  RETURN VALUES VAR.O_ID ;

END
%


--
-- ORDER STATUS
--

CREATE FUNCTION ORD_C_LAST(   W_ID INTEGER
                            , D_ID SMALLINT
                            , C_LAST VARCHAR(16)
```

```
                    )

RETURNS TABLE(   O_ID        INTEGER
               , O_CARRIER_ID SMALLINT
               , O_ENTRY_D    TIMESTAMP
               , C_BALANCE    DECIMAL(12,2)
               , C_FIRST      VARCHAR(16)
               , C_MIDDLE     CHAR(2)
               , C_ID         INTEGER
             )

SPECIFIC ORD_C_LAST

READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE C_BALANCE    DECIMAL(12,2) ;
  DECLARE C_FIRST      VARCHAR(16) ;
  DECLARE C_MIDDLE     CHAR(2) ;
  DECLARE C_ID         INTEGER ;
  DECLARE O_ID         INTEGER;
  DECLARE O_CARRIER_ID SMALLINT;
  DECLARE O_ENTRY_D    TIMESTAMP;

  /* Retrieve the Customer information */

  SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_ID )

    = ( SELECT  C_BALANCE, C_FIRST, C_MIDDLE , C_ID

        FROM ( SELECT   C_ID
                      , C_BALANCE
                      , C_FIRST
                      , C_MIDDLE
                      , COUNT(*) OVER() AS COUNT
                      , ROWNUMBER() OVER (ORDER BY C_FIRST) AS NUM

               FROM CUSTOMER
              WHERE C_W_ID = ORD_C_LAST.W_ID
                AND C_D_ID = ORD_C_LAST.D_ID
                AND C_LAST = ORD_C_LAST.C_LAST

             ) AS V1

       WHERE NUM = (COUNT + BIGINT( 1 ) ) / BIGINT( 2 )
      )
  ;

  SET ( O_ID , O_CARRIER_ID , O_ENTRY_D )
    =   ( SELECT   O_ID
                 , O_CARRIER_ID
                 , O_ENTRY_D

          FROM ORDERS

         WHERE O_W_ID = ORD_C_LAST.W_ID
           AND O_D_ID = ORD_C_LAST.D_ID
           AND O_C_ID = VAR.C_ID

         ORDER BY O_ID DESC
         FETCH FIRST 1 ROW ONLY

        )
  ;


  RETURN VALUES (   VAR.O_ID
                  , VAR.O_CARRIER_ID
                  , VAR.O_ENTRY_D
                  , VAR.C_BALANCE
                  , VAR.C_FIRST
                  , VAR.C_MIDDLE
                  , VAR.C_ID
                )
  ;


END
%

CREATE FUNCTION ORD_C_ID(   W_ID INTEGER
                          , D_ID SMALLINT
                          , C_ID INTEGER
                        )

RETURNS TABLE(   O_ID        INTEGER
               , O_CARRIER_ID SMALLINT
               , O_ENTRY_D    TIMESTAMP
               , C_BALANCE    DECIMAL(12,2)
               , C_FIRST      VARCHAR(16)
               , C_MIDDLE     CHAR(2)
               , C_LAST       VARCHAR(16)
             )

SPECIFIC ORD_C_ID

READS SQL DATA NO EXTERNAL ACTION DETERMINISTIC LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE C_BALANCE    DECIMAL(12,2) ;
  DECLARE C_FIRST      VARCHAR(16) ;
  DECLARE C_MIDDLE     CHAR(2) ;
  DECLARE C_LAST       VARCHAR(16) ;
  DECLARE O_ID         INTEGER;
  DECLARE O_CARRIER_ID SMALLINT;
  DECLARE O_ENTRY_D    TIMESTAMP;

  /* Retrieve the Customer information */

  SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_LAST )

    = ( SELECT  C_BALANCE, C_FIRST, C_MIDDLE, C_LAST

          FROM CUSTOMER

         WHERE C_ID   = ORD_C_ID.C_ID
           AND C_W_ID = ORD_C_ID.W_ID
           AND C_D_ID = ORD_C_ID.D_ID
      )
  ;

  SET (O_ID, O_CARRIER_ID, O_ENTRY_D)

    =   ( SELECT   O_ID
                 , O_CARRIER_ID
                 , O_ENTRY_D

          FROM ORDERS

              WHERE O_W_ID = ORD_C_ID.W_ID
                AND O_D_ID = ORD_C_ID.D_ID
                AND O_C_ID = ORD_C_ID.C_ID

         ORDER BY O_ID DESC
         FETCH FIRST 1 ROW ONLY
       )
  ;

  RETURN VALUES (   VAR.O_ID
                  , VAR.O_CARRIER_ID
                  , VAR.O_ENTRY_D
                  , VAR.C_BALANCE
                  , VAR.C_FIRST
                  , VAR.C_MIDDLE
                  , VAR.C_LAST
                ) ;

END
%



--
-- PAYMENT
--

CREATE FUNCTION PAY_C_LAST(  W_ID INTEGER
                           , D_ID SMALLINT
                           , C_W_ID INTEGER
                           , C_D_ID SMALLINT
                           , C_LAST VARCHAR(16)
                           , H_AMOUNT DECIMAL(6,2)
                           , BAD_CREDIT_PREFIX VARCHAR(28)
                          )

RETURNS TABLE(   W_STREET_1 CHAR(20)
               , W_STREET_2 CHAR(20)
               , W_CITY CHAR(20)
               , W_STATE CHAR(2)
               , W_ZIP CHAR(9)
               , D_STREET_1 CHAR(20)
               , D_STREET_2 CHAR(20)
               , D_CITY CHAR(20)
               , D_STATE CHAR(2)
               , D_ZIP CHAR(9)
               , C_ID INTEGER
               , C_FIRST VARCHAR(16)
               , C_MIDDLE CHAR(2)
               , C_STREET_1 VARCHAR(20)
               , C_STREET_2 VARCHAR(20)
               , C_CITY VARCHAR(20)
               , C_STATE CHAR(2)
               , C_ZIP CHAR(9)
               , C_PHONE CHAR(16)
               , C_SINCE TIMESTAMP
               , C_CREDIT CHAR(2)
               , C_CREDIT_LIM DECIMAL(12,2)
               , C_DISCOUNT INTEGER
               , C_BALANCE DECIMAL(12,2)
               , C_DATA CHAR(200)
               , H_DATE TIMESTAMP
             )

SPECIFIC PAY_C_LAST

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE  W_NAME CHAR(10) ;
  DECLARE  D_NAME CHAR(10) ;

  DECLARE  W_STREET_1 CHAR(20) ;
  DECLARE  W_STREET_2 CHAR(20) ;
  DECLARE  W_CITY CHAR(20) ;
  DECLARE  W_STATE CHAR(2) ;
  DECLARE  W_ZIP CHAR(9) ;
```

```
DECLARE  D_STREET_1 CHAR(20) ;
DECLARE  D_STREET_2 CHAR(20) ;
DECLARE  D_CITY CHAR(20) ;
DECLARE  D_STATE CHAR(2) ;
DECLARE  D_ZIP CHAR(9) ;

DECLARE  C_ID INTEGER ;

DECLARE  C_FIRST VARCHAR(16) ;
DECLARE  C_MIDDLE CHAR(2) ;
DECLARE  C_STREET_1 VARCHAR(20) ;
DECLARE  C_STREET_2 VARCHAR(20) ;
DECLARE  C_CITY VARCHAR(20) ;
DECLARE  C_STATE CHAR(2) ;
DECLARE  C_ZIP CHAR(9) ;
DECLARE  C_PHONE CHAR(16) ;
DECLARE  C_SINCE TIMESTAMP ;
DECLARE  C_CREDIT CHAR(2) ;
DECLARE  C_CREDIT_LIM DECIMAL(12,2) ;
DECLARE  C_DISCOUNT INTEGER ;
DECLARE  C_BALANCE DECIMAL(12,2) ;
DECLARE  C_DATA CHAR(200) ;

DECLARE  H_DATE TIMESTAMP ;

/* Generate the current date and time for the payment date */
SET H_DATE = CURRENT TIMESTAMP ;

/* Update District and retieve its data */

SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP)

  =  ( SELECT   D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP

       FROM OLD TABLE ( UPDATE DISTRICT

                   SET D_YTD = D_YTD + PAY_C_LAST.H_AMOUNT

                 WHERE D_W_ID = PAY_C_LAST.W_ID
                   AND D_ID   = PAY_C_LAST.D_ID
                ) AS U
    )
;

/* Determine the C_ID */

SET ( C_ID )

  = ( SELECT C_ID
      FROM ( SELECT   C_ID
                   , COUNT(*) OVER() AS COUNT
                   , ROWNUMBER() OVER (ORDER BY C_FIRST) AS NUM

             FROM CUSTOMER
             WHERE C_LAST = PAY_C_LAST.C_LAST
               AND C_W_ID = PAY_C_LAST.C_W_ID
               AND C_D_ID = PAY_C_LAST.C_D_ID
           ) AS T

     WHERE NUM = (COUNT + BIGINT( 1 ) ) / BIGINT( 2 )
   )
;


/* Update the middle customer */

SET (   C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
      , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
      , C_DISCOUNT, C_BALANCE, C_DATA )

   = ( SELECT   C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
             , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
             , C_DISCOUNT, C_BALANCE
             , CASE WHEN C_CREDIT = 'BC' THEN SUBSTR(C_DATA, 1, 200) ELSE NULL
END AS C_DATA

       FROM NEW TABLE ( UPDATE CUSTOMER

                     SET   C_BALANCE     = C_BALANCE - PAY_C_LAST.H_AMOUNT
                         , C_YTD_PAYMENT = C_YTD_PAYMENT +
PAY_C_LAST.H_AMOUNT
                         , C_PAYMENT_CNT = C_PAYMENT_CNT + SMALLINT( 1 )

                         , C_DATA = CASE WHEN C_CREDIT = 'BC'
                                     THEN    CHAR( C_ID )            -- 11 bytes long
                                          || BAD_CREDIT_PREFIX       -- 28 bytes long
                                          || SUBSTR( C_DATA, 1, 461 ) -- 461 + 39 = 500
                                     ELSE C_DATA
                                   END

                   WHERE C_W_ID = PAY_C_LAST.C_W_ID
                     AND C_D_ID = PAY_C_LAST.C_D_ID
                     AND C_ID   = VAR.C_ID
                  ) AS U
    )
;

/* Update the warehouse */

SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP )

 = ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP

       FROM OLD TABLE ( UPDATE WAREHOUSE

                     SET W_YTD = W_YTD + PAY_C_LAST.H_AMOUNT

                   WHERE W_ID = PAY_C_LAST.W_ID
                  ) AS U
    )
;

/* Finally insert into the warehouse */

INSERT

   INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID, H_W_ID, H_DATA,
H_DATE, H_AMOUNT )

   VALUES (   VAR.C_ID
            , PAY_C_LAST.C_D_ID
            , PAY_C_LAST.C_W_ID
            , PAY_C_LAST.D_ID
            , PAY_C_LAST.W_ID
            , VAR.W_NAME || CHAR( '    ', 4 )  || VAR.D_NAME
            , VAR.H_DATE
            , PAY_C_LAST.H_AMOUNT
          )
;

/* Done - return the collected data */

RETURN VALUES (   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
                , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
                , C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
                , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT,
C_CREDIT_LIM
                , C_DISCOUNT, C_BALANCE, C_DATA, H_DATE
              )
;
END
%

CREATE FUNCTION PAY_C_ID(  W_ID INTEGER
                         , D_ID SMALLINT
                         , C_W_ID INTEGER
                         , C_D_ID SMALLINT
                         , C_ID INTEGER
                         , H_AMOUNT DECIMAL(6,2)
                         , BAD_CREDIT_PREFIX VARCHAR(34)
                        )

RETURNS TABLE(   W_STREET_1 CHAR(20)
               , W_STREET_2 CHAR(20)
               , W_CITY CHAR(20)
               , W_STATE CHAR(2)
               , W_ZIP CHAR(9)
               , D_STREET_1 CHAR(20)
               , D_STREET_2 CHAR(20)
               , D_CITY CHAR(20)
               , D_STATE CHAR(2)
               , D_ZIP CHAR(9)
               , C_LAST VARCHAR(16)
               , C_FIRST VARCHAR(16)
               , C_MIDDLE CHAR(2)
               , C_STREET_1 VARCHAR(20)
               , C_STREET_2 VARCHAR(20)
               , C_CITY VARCHAR(20)
               , C_STATE CHAR(2)
               , C_ZIP CHAR(9)
               , C_PHONE CHAR(16)
               , C_SINCE TIMESTAMP
               , C_CREDIT CHAR(2)
               , C_CREDIT_LIM DECIMAL(12,2)
               , C_DISCOUNT REAL
               , C_BALANCE DECIMAL(12,2)
               , C_DATA CHAR(200)
               , H_DATE TIMESTAMP
             )
SPECIFIC PAY_C_ID

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE  W_NAME CHAR(10) ;
  DECLARE  D_NAME CHAR(10) ;

  DECLARE  W_STREET_1 CHAR(20) ;
  DECLARE  W_STREET_2 CHAR(20) ;
  DECLARE  W_CITY CHAR(20) ;
  DECLARE  W_STATE CHAR(2) ;
  DECLARE  W_ZIP CHAR(9) ;

  DECLARE  D_STREET_1 CHAR(20) ;
  DECLARE  D_STREET_2 CHAR(20) ;
  DECLARE  D_CITY CHAR(20) ;
  DECLARE  D_STATE CHAR(2) ;
  DECLARE  D_ZIP CHAR(9) ;

  DECLARE  C_LAST VARCHAR(16) ;

  DECLARE  C_FIRST VARCHAR(16) ;
  DECLARE  C_MIDDLE CHAR(2) ;
  DECLARE  C_STREET_1 VARCHAR(20) ;
  DECLARE  C_STREET_2 VARCHAR(20) ;
  DECLARE  C_CITY VARCHAR(20) ;
  DECLARE  C_STATE CHAR(2) ;
  DECLARE  C_ZIP CHAR(9) ;
```

```
  DECLARE  C_PHONE CHAR(16) ;
  DECLARE  C_SINCE TIMESTAMP ;
  DECLARE  C_CREDIT CHAR(2) ;
  DECLARE  C_CREDIT_LIM DECIMAL(12,2) ;
  DECLARE  C_DISCOUNT REAL ;
  DECLARE  C_BALANCE DECIMAL(12,2) ;
  DECLARE  C_DATA CHAR(200) ;
  DECLARE  H_DATE TIMESTAMP;

  /* Generate the current date and time for the payment date */
  SET H_DATE = CURRENT TIMESTAMP;

  /* Update District and retieve its data */

  SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP )

   =  ( SELECT   D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP

        FROM OLD TABLE ( UPDATE DISTRICT

                         SET D_YTD = D_YTD + PAY_C_ID.H_AMOUNT

                       WHERE D_W_ID = PAY_C_ID.W_ID
                         AND D_ID   = PAY_C_ID.D_ID
                     ) AS U
     )
  ;

  /* Update the middle customer */

  SET (   C_LAST, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
       , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
       , C_DISCOUNT, C_BALANCE, C_DATA )

   = ( SELECT   C_LAST, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
             , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
             , C_DISCOUNT, C_BALANCE
             , CASE WHEN C_CREDIT = 'BC' THEN SUBSTR(C_DATA, 1, 200) ELSE NULL
END AS C_DATA

         FROM NEW TABLE ( UPDATE CUSTOMER

                          SET   C_BALANCE     = C_BALANCE - PAY_C_ID.H_AMOUNT
                              , C_YTD_PAYMENT = C_YTD_PAYMENT + PAY_C_ID.H_AMOUNT
                              , C_PAYMENT_CNT = C_PAYMENT_CNT + SMALLINT( 1 )

                              , C_DATA = CASE WHEN C_CREDIT = 'BC'
                                         THEN    BAD_CREDIT_PREFIX        -- 34 bytes long
                                            || SUBSTR( C_DATA, 1, 466 ) -- 466 + 34 = 500 bytes
                                         ELSE C_DATA
                                       END

                        WHERE C_W_ID = PAY_C_ID.C_W_ID
                          AND C_D_ID = PAY_C_ID.C_D_ID
                          AND C_ID   = PAY_C_ID.C_ID
                      ) AS U
     )
  ;

  /* Update the warehouse */

  SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP )

   = ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP

         FROM OLD TABLE ( UPDATE WAREHOUSE

                          SET W_YTD = W_YTD + PAY_C_ID.H_AMOUNT

                        WHERE W_ID = PAY_C_ID.W_ID
                      ) AS U
     )
  ;

  /* Finally insert into the warehouse */

  INSERT

    INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID, H_W_ID, H_DATA,
H_DATE, H_AMOUNT )

    VALUES (   PAY_C_ID.C_ID
             , PAY_C_ID.C_D_ID
             , PAY_C_ID.C_W_ID
             , PAY_C_ID.D_ID
             , PAY_C_ID.W_ID
             , VAR.W_NAME || CHAR( '    ', 4 ) || VAR.D_NAME
             , VAR.H_DATE
             , PAY_C_ID.H_AMOUNT
           )
  ;

  /* Done - return the collected data */

  RETURN VALUES (   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
                  , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
                  , C_LAST, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
                  , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT,
C_CREDIT_LIM
                  , C_DISCOUNT, C_BALANCE, C_DATA, H_DATE
                )
  ;

END
%

--
-- NEW ORDER
--

CREATE FUNCTION NEW_OL_ALL(   I_ID      INT
                            , I_QTY     SMALLINT
                            , W_ID      INT
                            , SUPP_W_ID INT
                            , O_ID      INT
                            , D_ID      SMALLINT
                          )

RETURNS TABLE(   I_PRICE      DECIMAL(5,2)
               , I_NAME       CHAR(24)
               , I_DATA       VARCHAR(50)
               , OL_DIST_INFO  CHAR(24)
               , S_DATA       VARCHAR(50)
               , S_QUANTITY    SMALLINT
             )

SPECIFIC NEW_OL_ALL

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE I_PRICE       DECIMAL(5,2) ;
  DECLARE I_NAME        CHAR(24)  ;
  DECLARE I_DATA        VARCHAR(50)  ;
  DECLARE OL_DIST_INFO  CHAR(24)  ;
  DECLARE S_DATA        VARCHAR(50)  ;
  DECLARE S_QUANTITY    SMALLINT  ;

  SET   ( I_PRICE , I_NAME , I_DATA )

    =  ( SELECT
                I_PRICE
              , I_NAME
              , I_DATA

           FROM ITEM

          WHERE ITEM.I_ID = NEW_OL_ALL.I_ID
       ) ;

    SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )

     = ( SELECT   OL_DIST_INFO
               , S_DATA
               , S_QUANTITY

           FROM NEW TABLE (    UPDATE STOCK

                          INCLUDE (  OL_DIST_INFO CHAR( 24 )  )

                            SET S_QUANTITY = CASE WHEN S_QUANTITY -
NEW_OL_ALL.I_QTY >= 10
                                                  THEN S_QUANTITY - NEW_OL_ALL.I_QTY
                                                  ELSE S_QUANTITY - NEW_OL_ALL.I_QTY + 91
                                                END

                                , S_ORDER_CNT = S_ORDER_CNT + SMALLINT( 1 )

                                , S_YTD = S_YTD + NEW_OL_ALL.I_QTY

                                , S_REMOTE_CNT = CASE WHEN
NEW_OL_ALL.SUPP_W_ID = NEW_OL_ALL.W_ID
                                                      THEN S_REMOTE_CNT
                                                      ELSE S_REMOTE_CNT + SMALLINT( 1 )
                                                    END

                                , OL_DIST_INFO = CASE D_ID WHEN SMALLINT(  1 )
THEN S_DIST_01
                                                           WHEN SMALLINT(  2 )  THEN
S_DIST_02
                                                           WHEN SMALLINT(  3 )  THEN
S_DIST_03
                                                           WHEN SMALLINT(  4 )  THEN
S_DIST_04
                                                           WHEN SMALLINT(  5 )  THEN
S_DIST_05
                                                           WHEN SMALLINT(  6 )  THEN
S_DIST_06
                                                           WHEN SMALLINT(  7 )  THEN
S_DIST_07
                                                           WHEN SMALLINT(  8 )  THEN
S_DIST_08
                                                           WHEN SMALLINT(  9 )  THEN
S_DIST_09
                                                           WHEN SMALLINT( 10 )  THEN
S_DIST_10
                                                         END
                              WHERE S_I_ID = NEW_OL_ALL.I_ID
                                AND S_W_ID = NEW_OL_ALL.SUPP_W_ID
                            ) AS U
       )
    ;

    RETURN VALUES(   VAR.I_PRICE
                   , VAR.I_NAME
                   , VAR.I_DATA
                   , VAR.OL_DIST_INFO
                   , VAR.S_DATA
```

```
              , VAR.S_QUANTITY
            )
  ;

END
%

CREATE FUNCTION NEW_OL_LOCAL(   I_ID      INT
                        , I_QTY     SMALLINT
                        , W_ID      INT
                        , O_ID      INT
                        , D_ID      SMALLINT
                       )

RETURNS TABLE(   I_PRICE       DECIMAL(5,2)
            , I_NAME        CHAR(24)
            , I_DATA        VARCHAR(50)
            , OL_DIST_INFO  CHAR(24)
            , S_DATA        VARCHAR(50)
            , S_QUANTITY    SMALLINT
          )

SPECIFIC NEW_OL_LOCAL

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE I_PRICE       DECIMAL(5,2)  ;
 DECLARE I_NAME        CHAR(24)  ;
 DECLARE I_DATA        VARCHAR(50)  ;
 DECLARE OL_DIST_INFO  CHAR(24)  ;
 DECLARE S_DATA        VARCHAR(50)  ;
 DECLARE S_QUANTITY    SMALLINT  ;


 SET   ( I_PRICE , I_NAME , I_DATA )

  =  ( SELECT
            I_PRICE
          , I_NAME
          , I_DATA

       FROM ITEM

      WHERE ITEM.I_ID = NEW_OL_LOCAL.I_ID
    ) ;

  SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )

   = ( SELECT   OL_DIST_INFO
          , S_DATA
          , S_QUANTITY

       FROM NEW TABLE (    UPDATE STOCK

                    INCLUDE (  OL_DIST_INFO CHAR( 24 )  )

                      SET S_QUANTITY = CASE WHEN S_QUANTITY -
NEW_OL_LOCAL.I_QTY >= 10
                                       THEN S_QUANTITY - NEW_OL_LOCAL.I_QTY
                                       ELSE S_QUANTITY - NEW_OL_LOCAL.I_QTY + 91
                                  END

                                  , S_ORDER_CNT = S_ORDER_CNT + SMALLINT( 1 )

                                  , S_YTD = S_YTD + NEW_OL_LOCAL.I_QTY

                                  , OL_DIST_INFO = CASE D_ID WHEN SMALLINT(  1 )
THEN S_DIST_01
                                                     WHEN SMALLINT(  2 )  THEN
S_DIST_02
                                                     WHEN SMALLINT(  3 )  THEN
S_DIST_03
                                                     WHEN SMALLINT(  4 )  THEN
S_DIST_04
                                                     WHEN SMALLINT(  5 )  THEN
S_DIST_05
                                                     WHEN SMALLINT(  6 )  THEN
S_DIST_06
                                                     WHEN SMALLINT(  7 )  THEN
S_DIST_07
                                                     WHEN SMALLINT(  8 )  THEN
S_DIST_08
                                                     WHEN SMALLINT(  9 )  THEN
S_DIST_09
                                                     WHEN SMALLINT( 10 )  THEN
S_DIST_10
                                                END
                      WHERE S_I_ID = NEW_OL_LOCAL.I_ID
                        AND S_W_ID = NEW_OL_LOCAL.W_ID
                    ) AS U
     )
   ;

   RETURN VALUES(   VAR.I_PRICE
              , VAR.I_NAME
              , VAR.I_DATA
              , VAR.OL_DIST_INFO
              , VAR.S_DATA
              , VAR.S_QUANTITY
             )
   ;

END
%

CREATE FUNCTION NEW_WH (   O_ID       INTEGER
                  , W_ID       INTEGER
                  , D_ID       SMALLINT
                  , C_ID       INTEGER
                  , O_OL_CNT    SMALLINT
                  , O_ALL_LOCAL SMALLINT
                 )

RETURNS TABLE (   W_TAX       REAL
            , C_DISCOUNT  REAL
            , C_LAST      VARCHAR(16)
            , C_CREDIT    CHAR(2)
            , O_ENTRY_D   TIMESTAMP
           )

SPECIFIC NEW_WH

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE C_DISCOUNT  REAL ;
 DECLARE C_LAST      VARCHAR(16) ;
 DECLARE C_CREDIT    CHAR(2) ;
 DECLARE W_TAX       REAL ;
 DECLARE O_ENTRY_D   TIMESTAMP;

 SET O_ENTRY_D = CURRENT TIMESTAMP;

 INSERT

  INTO NEW_ORDER ( NO_O_ID, NO_D_ID, NO_W_ID )

         VALUES (   O_ID
                , D_ID
                , W_ID
               )
 ;


 INSERT

   INTO ORDERS ( O_C_ID, O_ENTRY_D, O_CARRIER_ID, O_OL_CNT, O_ALL_LOCAL
O_ID, O_W_ID, O_D_ID )

       VALUES (   C_ID
              , O_ENTRY_D
              , 0
              , O_OL_CNT
              , O_ALL_LOCAL
              , O_ID
              , W_ID
              , D_ID
             )
 ;

 SET ( C_DISCOUNT, C_LAST, C_CREDIT )

  =  ( SELECT  C_DISCOUNT, C_LAST, C_CREDIT

       FROM  CUSTOMER

      WHERE  C_ID = NEW_WH.C_ID
        AND  C_W_ID = W_ID
        AND  C_D_ID = D_ID
    )
 ;


 SET W_TAX
  =    ( SELECT W_TAX

         FROM WAREHOUSE

        WHERE W_ID = NEW_WH.W_ID
      )
 ;

 RETURN VALUES ( W_TAX , C_DISCOUNT , C_LAST , C_CREDIT, O_ENTRY_D ) ;

END
%
```

Src.Srv/cat-proc.ddl

```
CREATE PROCEDURE news
                (in new_in   varchar(262) FOR BIT DATA,
                 out new_out varchar(682) FOR BIT DATA)
                LANGUAGE C
                PARAMETER STYLE GENERAL
                EXTERNAL NAME '/home/tpcc/sqllib/function/news!news'
                not fenced;

CREATE PROCEDURE ords
                (in ord_in   varchar(42) FOR BIT DATA,
                 out ord_out varchar(822) FOR BIT DATA)
                LANGUAGE C
                PARAMETER STYLE GENERAL
                EXTERNAL NAME '/home/tpcc/sqllib/function/ords!ords'
                not fenced;

CREATE PROCEDURE dels
                (in del_in   varchar(14) FOR BIT DATA,
```

```
 out del_out varchar(50) FOR BIT DATA)
LANGUAGE C
PARAMETER STYLE GENERAL
EXTERNAL NAME '/home/tpcc/sqllib/function/dels!dels'
not fenced;
```

Src.Srv/tpcc_all_sql.sqc

```
/************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
************************************************************************/

/*
 * tpcc_all_sql.sqc - Client/Server code for TPCC
 *
 */

#include <stdlib.h>
#include <errno.h>
#include "db2tpcc.h"
#include "tpccapp.h"
#include "tpccdbg.h"

#include "sqlca.h"
#include "sql.h"

// ---------------------------------------------------------------
// New Order  SERVER
// ---------------------------------------------------------------

int  static is_ORIGINAL( char *string, short length ) ;

SQL_API_RC new_order_internal( char *pin, char *pout )
{

   struct out_neword_struct *neword;

   struct in_neword_struct  *in_neword;

   struct sqlca sqlca ;

   int     fbadItemDetected = 0 ;

   EXEC SQL BEGIN DECLARE SECTION;

      char    c_last [ 16 ] ;
      char    c_credit [ 2 ] ;
      float  c_discount ;
      float  dist_tax ;
      float  ware_tax ;

      sqlint32   w_id ;
      short      d_id ;
      sqlint32   c_id ;

      sqlint32   next_o_id ;

      short      s_quantity ;

      sqlint32   supply_w_id ;

   short   inputItemCount ;

   char    stockDistrictInformation [ 24 ];
   char    item_name[ 24 ] ;

   char    o_entry_d [27];

   short   allLocal ;

   float    item_price ;

   struct i_data_type { short len ; char data[ 50 ] ; } i_data ;
   struct s_data_type { short len ; char data[ 50 ] ; } s_data ;

   sqlint32    id0, id1, id2,  id3,  id4,  id5,  id6,  id7;
   sqlint32    id8, id9, id10, id11, id12, id13, id14;

   sqlint32   supply_w_id0,  supply_w_id1, supply_w_id2,  supply_w_id3;
   sqlint32   supply_w_id4,  supply_w_id5, supply_w_id6,  supply_w_id7;
   sqlint32   supply_w_id8,  supply_w_id9, supply_w_id10,  supply_w_id11;
   sqlint32   supply_w_id12,  supply_w_id13, supply_w_id14;

   short   ol_quantity0,  ol_quantity1, ol_quantity2,  ol_quantity3;
   short   ol_quantity4,  ol_quantity5, ol_quantity6,  ol_quantity7;
   short   ol_quantity8,  ol_quantity9, ol_quantity10,  ol_quantity11;
   short   ol_quantity12,  ol_quantity13, ol_quantity14;


EXEC SQL END DECLARE SECTION;

int      storedProcRc ;
int      inputItemArrayIndex ;

char    stockDistrictInformationArray [15][25];

#define stockDistrictInformation  stockDistrictInformationArray[ inputItemArrayIndex ]

// Redirected input fields

#define w_id in_neword->s_W_ID
#define d_id in_neword->s_D_ID
#define c_id in_neword->s_C_ID

#define inputItemCount in_neword->s_O_OL_CNT

#define allLocal in_neword->s_all_local

// Redirected output fields

#define  c_last       neword->s_C_LAST
#define  c_credit     neword->s_C_CREDIT
#define  c_discount   neword->s_C_DISCOUNT
#define  ware_tax     neword->s_W_TAX
#define  dist_tax     neword->s_D_TAX
#define  s_quantity   neword->item[ inputItemArrayIndex ].s_S_QUANTITY
#define o_entry_d     neword->s_O_ENTRY_D_time

// This output field becomes an input field to order_line

#define next_o_id neword->s_O_ID

// item price/name

#define  item_name    neword->item[ inputItemArrayIndex ].s_I_NAME

float    i_priceArray[ 15 ] ;

#define  item_price   i_priceArray[ inputItemArrayIndex ]

// Handle the generic/brand distinction

struct  i_data_type i_dataArray[ 15 ] ;
struct  s_data_type s_dataArray[ 15 ] ;

#define  i_data       i_dataArray[ inputItemArrayIndex ]
#define  s_data       s_dataArray[ inputItemArrayIndex ]

// Redirect hostvars to input structure

#define  id0  in_neword->in_item[0].s_OL_I_ID
#define  id1  in_neword->in_item[1].s_OL_I_ID
#define  id2  in_neword->in_item[2].s_OL_I_ID
#define  id3  in_neword->in_item[3].s_OL_I_ID
#define  id4  in_neword->in_item[4].s_OL_I_ID
#define  id5  in_neword->in_item[5].s_OL_I_ID
#define  id6  in_neword->in_item[6].s_OL_I_ID
#define  id7  in_neword->in_item[7].s_OL_I_ID
#define  id8  in_neword->in_item[8].s_OL_I_ID
#define  id9  in_neword->in_item[9].s_OL_I_ID
#define  id10 in_neword->in_item[10].s_OL_I_ID
#define  id11 in_neword->in_item[11].s_OL_I_ID
#define  id12 in_neword->in_item[12].s_OL_I_ID
#define  id13 in_neword->in_item[13].s_OL_I_ID
#define  id14 in_neword->in_item[14].s_OL_I_ID

#define ol_quantity0  in_neword->in_item[  0 ].s_OL_QUANTITY
#define ol_quantity1  in_neword->in_item[  1 ].s_OL_QUANTITY
#define ol_quantity2  in_neword->in_item[  2 ].s_OL_QUANTITY
#define ol_quantity3  in_neword->in_item[  3 ].s_OL_QUANTITY
#define ol_quantity4  in_neword->in_item[  4 ].s_OL_QUANTITY
#define ol_quantity5  in_neword->in_item[  5 ].s_OL_QUANTITY
#define ol_quantity6  in_neword->in_item[  6 ].s_OL_QUANTITY
#define ol_quantity7  in_neword->in_item[  7 ].s_OL_QUANTITY
#define ol_quantity8  in_neword->in_item[  8 ].s_OL_QUANTITY
#define ol_quantity9  in_neword->in_item[  9 ].s_OL_QUANTITY
#define ol_quantity10 in_neword->in_item[ 10 ].s_OL_QUANTITY
#define ol_quantity11 in_neword->in_item[ 11 ].s_OL_QUANTITY
#define ol_quantity12 in_neword->in_item[ 12 ].s_OL_QUANTITY
#define ol_quantity13 in_neword->in_item[ 13 ].s_OL_QUANTITY
#define ol_quantity14 in_neword->in_item[ 14 ].s_OL_QUANTITY

#define supply_w_id0  in_neword->in_item[  0 ].s_OL_SUPPLY_W_ID
#define supply_w_id1  in_neword->in_item[  1 ].s_OL_SUPPLY_W_ID
#define supply_w_id2  in_neword->in_item[  2 ].s_OL_SUPPLY_W_ID
#define supply_w_id3  in_neword->in_item[  3 ].s_OL_SUPPLY_W_ID
#define supply_w_id4  in_neword->in_item[  4 ].s_OL_SUPPLY_W_ID
#define supply_w_id5  in_neword->in_item[  5 ].s_OL_SUPPLY_W_ID
#define supply_w_id6  in_neword->in_item[  6 ].s_OL_SUPPLY_W_ID
#define supply_w_id7  in_neword->in_item[  7 ].s_OL_SUPPLY_W_ID
#define supply_w_id8  in_neword->in_item[  8 ].s_OL_SUPPLY_W_ID
#define supply_w_id9  in_neword->in_item[  9 ].s_OL_SUPPLY_W_ID
#define supply_w_id10 in_neword->in_item[ 10 ].s_OL_SUPPLY_W_ID
#define supply_w_id11 in_neword->in_item[ 11 ].s_OL_SUPPLY_W_ID
#define supply_w_id12 in_neword->in_item[ 12 ].s_OL_SUPPLY_W_ID
#define supply_w_id13 in_neword->in_item[ 13 ].s_OL_SUPPLY_W_ID
#define supply_w_id14 in_neword->in_item[ 14 ].s_OL_SUPPLY_W_ID

EXEC SQL DECLARE ISOL_Remote_1 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
```

```
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                            , :w_id AS W_ID
                            , :d_id as D_ID
                            , OL_NUMBER
                            , I_ID
                            , I_SUPPLY_W_ID
                            , I_QTY

                        FROM  Table( VALUES


(  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                        ) AS ITEMLIST

                        , TABLE( NEW_OL_ALL(   I_ID
                                    , I_QTY
                                    , W_ID
                                    , I_SUPPLY_W_ID
                                    , O_ID
                                    , D_ID
                                    )
                                ) AS NEW_OL_ALL

                WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
        )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
              , OL_W_ID
              , OL_NUMBER
              , OL_I_ID
              , OL_SUPPLY_W_ID
              , OL_DELIVERY_D
              , OL_QUANTITY
              , OL_AMOUNT
              , OL_DIST_INFO
            )

            INCLUDE (   I_PRICE    DECIMAL(5,2)
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )

                SELECT   O_ID
                      , D_ID
                      , W_ID
                      , OL_NUMBER
                      , I_ID
                      , I_SUPPLY_W_ID
                      , OL_DELIVERY_D
                      , I_QTY
                      , TOTAL_PRICE
                      , OL_DIST_INFO
                      , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM DATA

            ) AS INS
  ;
```

```
  EXEC SQL DECLARE ISOL_Remote_2 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                                FROM  Table( VALUES


(  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                    ,
(  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )

                                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                                ) AS ITEMLIST

                                , TABLE( NEW_OL_ALL(   I_ID
                                            , I_QTY
                                            , W_ID
                                            , I_SUPPLY_W_ID
                                            , O_ID
                                            , D_ID
                                            )
                                        ) AS NEW_OL_ALL

                        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
              , OL_W_ID
              , OL_NUMBER
              , OL_I_ID
              , OL_SUPPLY_W_ID
              , OL_DELIVERY_D
              , OL_QUANTITY
              , OL_AMOUNT
              , OL_DIST_INFO
            )

            INCLUDE (   I_PRICE    DECIMAL(5,2)
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )

                SELECT   O_ID
                      , D_ID
                      , W_ID
                      , OL_NUMBER
                      , I_ID
                      , I_SUPPLY_W_ID
                      , OL_DELIVERY_D
                      , I_QTY
                      , TOTAL_PRICE
                      , OL_DIST_INFO
                      , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM DATA

            ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_3 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                                FROM  Table( VALUES


(  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                    ,
(  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                    ,
(  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )

                                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                                ) AS ITEMLIST

                                , TABLE( NEW_OL_ALL(   I_ID
                                            , I_QTY
                                            , W_ID
                                            , I_SUPPLY_W_ID
                                            , O_ID
                                            , D_ID
                                            )
                                        ) AS NEW_OL_ALL

                        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
```

```
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
               )

               INCLUDE (   I_PRICE    DECIMAL(5,2)
                       , I_NAME     CHAR(24)
                       , I_DATA     VARCHAR(50)
                       , S_DATA     VARCHAR(50)
                       , S_QUANTITY SMALLINT )

                    SELECT   O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , OL_DELIVERY_D
                         , I_QTY
                         , TOTAL_PRICE
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                    FROM DATA

              ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_4 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
              , D_ID
              , W_ID
              , OL_NUMBER
              , I_ID
              , I_SUPPLY_W_ID
              , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
              , I_QTY
              , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
              , OL_DIST_INFO
              , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                    FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , I_QTY

                         FROM  Table( VALUES


(  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                              ,
(  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                              ,
(  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                              ,
(  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )

                                   ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID  )
                         ) AS ITEMLIST

                        , TABLE( NEW_OL_ALL(   I_ID
                                      , I_QTY
                                      , W_ID
                                      , I_SUPPLY_W_ID
                                      , O_ID
                                      , D_ID
                                     )
                               ) AS NEW_OL_ALL

                          WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                    )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

                (   OL_O_ID
                  , OL_D_ID
                  , OL_W_ID
                  , OL_NUMBER
                  , OL_I_ID
                  , OL_SUPPLY_W_ID
                  , OL_DELIVERY_D
                  , OL_QUANTITY
                  , OL_AMOUNT
                  , OL_DIST_INFO
                )

                INCLUDE (   I_PRICE    DECIMAL(5,2)
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )

                    SELECT   O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , OL_DELIVERY_D
                         , I_QTY
                         , TOTAL_PRICE
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                    FROM DATA

              ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_5 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                            , :w_id AS W_ID
                            , :d_id as D_ID
                            , OL_NUMBER
                            , I_ID
                            , I_SUPPLY_W_ID
                            , I_QTY

                       FROM  Table( VALUES


(  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                            ,
(  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                            ,
(  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                            ,
(  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                            ,
(  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )

                                 ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID  )
                       ) AS ITEMLIST

                      , TABLE( NEW_OL_ALL(   I_ID
                                    , I_QTY
                                    , W_ID
                                    , I_SUPPLY_W_ID
                                    , O_ID
                                    , D_ID
                                   )
                             ) AS NEW_OL_ALL

                        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                  )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

      FROM NEW TABLE ( INSERT INTO ORDER_LINE

                (   OL_O_ID
                  , OL_D_ID
                  , OL_W_ID
                  , OL_NUMBER
                  , OL_I_ID
                  , OL_SUPPLY_W_ID
                  , OL_DELIVERY_D
                  , OL_QUANTITY
                  , OL_AMOUNT
                  , OL_DIST_INFO
                )

                INCLUDE (   I_PRICE    DECIMAL(5,2)
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )

                    SELECT   O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , OL_DELIVERY_D
                         , I_QTY
                         , TOTAL_PRICE
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                    FROM DATA
```

```
                ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_6 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                                FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                                ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                                ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                                ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                                ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                                ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )

                                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                                ) AS ITEMLIST

                                , TABLE( NEW_OL_ALL(   I_ID
                                                    , I_QTY
                                                    , W_ID
                                                    , I_SUPPLY_W_ID
                                                    , O_ID
                                                    , D_ID
                                                    )
                                ) AS NEW_OL_ALL

                        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


      SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

        FROM NEW TABLE ( INSERT INTO ORDER_LINE

                    (   OL_O_ID
                      , OL_D_ID
                      , OL_W_ID
                      , OL_NUMBER
                      , OL_I_ID
                      , OL_SUPPLY_W_ID
                      , OL_DELIVERY_D
                      , OL_QUANTITY
                      , OL_AMOUNT
                      , OL_DIST_INFO
                    )

                    INCLUDE (   I_PRICE    DECIMAL(5,2)
                            , I_NAME     CHAR(24)
                            , I_DATA     VARCHAR(50)
                            , S_DATA     VARCHAR(50)
                            , S_QUANTITY SMALLINT )

                        SELECT   O_ID
                            , D_ID
                            , W_ID
                            , OL_NUMBER
                            , I_ID
                            , I_SUPPLY_W_ID
                            , OL_DELIVERY_D
                            , I_QTY
                            , TOTAL_PRICE
                            , OL_DIST_INFO
                            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM DATA

                    ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_7 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                                FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                                ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                                ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                                ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                                ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                                ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                                ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )

                                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                                ) AS ITEMLIST

                                , TABLE( NEW_OL_ALL(   I_ID
                                                    , I_QTY
                                                    , W_ID
                                                    , I_SUPPLY_W_ID
                                                    , O_ID
                                                    , D_ID
                                                    )
                                ) AS NEW_OL_ALL

                        WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


      SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

        FROM NEW TABLE ( INSERT INTO ORDER_LINE

                    (   OL_O_ID
                      , OL_D_ID
                      , OL_W_ID
                      , OL_NUMBER
                      , OL_I_ID
                      , OL_SUPPLY_W_ID
                      , OL_DELIVERY_D
                      , OL_QUANTITY
                      , OL_AMOUNT
                      , OL_DIST_INFO
                    )

                    INCLUDE (   I_PRICE    DECIMAL(5,2)
                            , I_NAME     CHAR(24)
                            , I_DATA     VARCHAR(50)
                            , S_DATA     VARCHAR(50)
                            , S_QUANTITY SMALLINT )

                        SELECT   O_ID
                            , D_ID
                            , W_ID
                            , OL_NUMBER
                            , I_ID
                            , I_SUPPLY_W_ID
                            , OL_DELIVERY_D
                            , I_QTY
                            , TOTAL_PRICE
                            , OL_DIST_INFO
                            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM DATA

                    ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_8 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
```

```
                        , I_SUPPLY_W_ID
                        , I_QTY

                    FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                        ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                        ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                        ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                        ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                        ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                        ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                        ,
( SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )

                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                    ) AS ITEMLIST

                    , TABLE( NEW_OL_ALL(   I_ID
                                , I_QTY
                                , W_ID
                                , I_SUPPLY_W_ID
                                , O_ID
                                , D_ID
                                )
                            ) AS NEW_OL_ALL

                WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
        )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
              , OL_W_ID
              , OL_NUMBER
              , OL_I_ID
              , OL_SUPPLY_W_ID
              , OL_DELIVERY_D
              , OL_QUANTITY
              , OL_AMOUNT
              , OL_DIST_INFO
            )

            INCLUDE (   I_PRICE    DECIMAL(5,2)
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )

                SELECT   O_ID
                    , D_ID
                    , W_ID
                    , OL_NUMBER
                    , I_ID
                    , I_SUPPLY_W_ID
                    , OL_DELIVERY_D
                    , I_QTY
                    , TOTAL_PRICE
                    , OL_DIST_INFO
                    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM DATA

            ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_9 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                        , :w_id AS W_ID
                        , :d_id as D_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , I_QTY

                    FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                        ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                        ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                        ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                        ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                        ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                        ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                        ,
( SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                        ,
( SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )

                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                    ) AS ITEMLIST

                    , TABLE( NEW_OL_ALL(   I_ID
                                , I_QTY
                                , W_ID
                                , I_SUPPLY_W_ID
                                , O_ID
                                , D_ID
                                )
                            ) AS NEW_OL_ALL

                WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
        )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
              , OL_W_ID
              , OL_NUMBER
              , OL_I_ID
              , OL_SUPPLY_W_ID
              , OL_DELIVERY_D
              , OL_QUANTITY
              , OL_AMOUNT
              , OL_DIST_INFO
            )

            INCLUDE (   I_PRICE    DECIMAL(5,2)
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )

                SELECT   O_ID
                    , D_ID
                    , W_ID
                    , OL_NUMBER
                    , I_ID
                    , I_SUPPLY_W_ID
                    , OL_DELIVERY_D
                    , I_QTY
                    , TOTAL_PRICE
                    , OL_DIST_INFO
                    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM DATA

            ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                        , :w_id AS W_ID
                        , :d_id as D_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , I_QTY

                    FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                        ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                        ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                        ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
```

```
                                ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                ,
( SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                ,
( SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                ,
( SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )

                              ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                            ) AS ITEMLIST

                           , TABLE( NEW_OL_ALL(   I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                       )
                                  ) AS NEW_OL_ALL

                          WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
              )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

                 (   OL_O_ID
                   , OL_D_ID
                   , OL_W_ID
                   , OL_NUMBER
                   , OL_I_ID
                   , OL_SUPPLY_W_ID
                   , OL_DELIVERY_D
                   , OL_QUANTITY
                   , OL_AMOUNT
                   , OL_DIST_INFO
                 )

                 INCLUDE (   I_PRICE    DECIMAL(5,2)
                         , I_NAME     CHAR(24)
                         , I_DATA     VARCHAR(50)
                         , S_DATA     VARCHAR(50)
                         , S_QUANTITY SMALLINT )

                        SELECT   O_ID
                              , D_ID
                              , W_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , OL_DELIVERY_D
                              , I_QTY
                              , TOTAL_PRICE
                              , OL_DIST_INFO
                              , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_11 CURSOR FOR

     WITH DATA AS (  SELECT  O_ID
                 , D_ID
                 , W_ID
                 , OL_NUMBER
                 , I_ID
                 , I_SUPPLY_W_ID
                 , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                 , I_QTY
                 , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                 , OL_DIST_INFO
                 , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                       FROM ( SELECT   :next_o_id as O_ID
                                 , :w_id AS W_ID
                                 , :d_id as D_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_SUPPLY_W_ID
                                 , I_QTY

                             FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                ,
( SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                ,
( SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                ,
( SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                ,
( SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10  )

                              ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                            ) AS ITEMLIST

                           , TABLE( NEW_OL_ALL(   I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                       )
                                  ) AS NEW_OL_ALL

                          WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
              )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

                 (   OL_O_ID
                   , OL_D_ID
                   , OL_W_ID
                   , OL_NUMBER
                   , OL_I_ID
                   , OL_SUPPLY_W_ID
                   , OL_DELIVERY_D
                   , OL_QUANTITY
                   , OL_AMOUNT
                   , OL_DIST_INFO
                 )

                 INCLUDE (   I_PRICE    DECIMAL(5,2)
                         , I_NAME     CHAR(24)
                         , I_DATA     VARCHAR(50)
                         , S_DATA     VARCHAR(50)
                         , S_QUANTITY SMALLINT )

                        SELECT   O_ID
                              , D_ID
                              , W_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , OL_DELIVERY_D
                              , I_QTY
                              , TOTAL_PRICE
                              , OL_DIST_INFO
                              , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_12 CURSOR FOR

     WITH DATA AS (  SELECT  O_ID
                 , D_ID
                 , W_ID
                 , OL_NUMBER
                 , I_ID
                 , I_SUPPLY_W_ID
                 , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                 , I_QTY
                 , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                 , OL_DIST_INFO
                 , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                       FROM ( SELECT   :next_o_id as O_ID
                                 , :w_id AS W_ID
                                 , :d_id as D_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_SUPPLY_W_ID
                                 , I_QTY

                             FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
```

```
                                ,
(  SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                ,
(  SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                ,
(  SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                ,
(  SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                                ,
(  SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )

                              ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                            ) AS ITEMLIST

                           , TABLE( NEW_OL_ALL(   I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                       )
                                  ) AS NEW_OL_ALL

                          WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

                  (   OL_O_ID
                    , OL_D_ID
                    , OL_W_ID
                    , OL_NUMBER
                    , OL_I_ID
                    , OL_SUPPLY_W_ID
                    , OL_DELIVERY_D
                    , OL_QUANTITY
                    , OL_AMOUNT
                    , OL_DIST_INFO
                  )

                  INCLUDE (   I_PRICE    DECIMAL(5,2)
                            , I_NAME     CHAR(24)
                            , I_DATA     VARCHAR(50)
                            , S_DATA     VARCHAR(50)
                            , S_QUANTITY SMALLINT )

                          SELECT   O_ID
                               , D_ID
                               , W_ID
                               , OL_NUMBER
                               , I_ID
                               , I_SUPPLY_W_ID
                               , OL_DELIVERY_D
                               , I_QTY
                               , TOTAL_PRICE
                               , OL_DIST_INFO
                               , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                          FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_13 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , I_SUPPLY_W_ID
                  , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                  , I_QTY
                  , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                               FROM  Table( VALUES


(  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                ,
(  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                ,
(  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                ,
(  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                ,
(  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                ,
(  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                ,
(  SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                ,
(  SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                ,
(  SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                ,
(  SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                ,
(  SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                                ,
(  SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )
                                ,
(  SMALLINT( 13 )      , :id12 , :ol_quantity12 , :supply_w_id12 )

                              ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                            ) AS ITEMLIST

                           , TABLE( NEW_OL_ALL(   I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                       )
                                  ) AS NEW_OL_ALL

                          WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

                  (   OL_O_ID
                    , OL_D_ID
                    , OL_W_ID
                    , OL_NUMBER
                    , OL_I_ID
                    , OL_SUPPLY_W_ID
                    , OL_DELIVERY_D
                    , OL_QUANTITY
                    , OL_AMOUNT
                    , OL_DIST_INFO
                  )

                  INCLUDE (   I_PRICE    DECIMAL(5,2)
                            , I_NAME     CHAR(24)
                            , I_DATA     VARCHAR(50)
                            , S_DATA     VARCHAR(50)
                            , S_QUANTITY SMALLINT )

                          SELECT   O_ID
                               , D_ID
                               , W_ID
                               , OL_NUMBER
                               , I_ID
                               , I_SUPPLY_W_ID
                               , OL_DELIVERY_D
                               , I_QTY
                               , TOTAL_PRICE
                               , OL_DIST_INFO
                               , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                          FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_14 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , I_SUPPLY_W_ID
                  , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                  , I_QTY
                  , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                               FROM  Table( VALUES


(  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                ,
(  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                ,
(  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                ,
(  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                ,
(  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                ,
(  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
```

```
                                ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                ,
( SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                ,
( SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                ,
( SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                ,
( SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                                ,
( SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )
                                ,
( SMALLINT( 13 )      , :id12 , :ol_quantity12 , :supply_w_id12 )
                                ,
( SMALLINT( 14 )      , :id13 , :ol_quantity13 , :supply_w_id13 )

                              ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                            ) AS ITEMLIST

                           , TABLE( NEW_OL_ALL(   I_ID
                                                , I_QTY
                                                , W_ID
                                                , I_SUPPLY_W_ID
                                                , O_ID
                                                , D_ID
                                                )
                                  ) AS NEW_OL_ALL

                         WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

                   (   OL_O_ID
                     , OL_D_ID
                     , OL_W_ID
                     , OL_NUMBER
                     , OL_I_ID
                     , OL_SUPPLY_W_ID
                     , OL_DELIVERY_D
                     , OL_QUANTITY
                     , OL_AMOUNT
                     , OL_DIST_INFO
                   )

                   INCLUDE (   I_PRICE    DECIMAL(5,2)
                             , I_NAME     CHAR(24)
                             , I_DATA     VARCHAR(50)
                             , S_DATA     VARCHAR(50)
                             , S_QUANTITY SMALLINT )

                          SELECT   O_ID
                                 , D_ID
                                 , W_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_SUPPLY_W_ID
                                 , OL_DELIVERY_D
                                 , I_QTY
                                 , TOTAL_PRICE
                                 , OL_DIST_INFO
                                 , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                          FROM DATA
                    ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Remote_15 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , I_SUPPLY_W_ID
                  , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                  , I_QTY
                  , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                     , :w_id AS W_ID
                                     , :d_id as D_ID
                                     , OL_NUMBER
                                     , I_ID
                                     , I_SUPPLY_W_ID
                                     , I_QTY

                               FROM  Table( VALUES


( SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                ,
( SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                ,
( SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                ,
( SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                ,
( SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                ,
( SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                ,
( SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                ,
( SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                ,
( SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                ,
( SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                ,
( SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                                ,
( SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )
                                ,
( SMALLINT( 13 )      , :id12 , :ol_quantity12 , :supply_w_id12 )
                                ,
( SMALLINT( 14 )      , :id13 , :ol_quantity13 , :supply_w_id13 )
                                ,
( SMALLINT( 15 )      , :id14 , :ol_quantity14 , :supply_w_id14 )

                              ) AS X ( OL_NUMBER , I_ID  , I_QTY         ,
I_SUPPLY_W_ID )
                            ) AS ITEMLIST

                           , TABLE( NEW_OL_ALL(   I_ID
                                                , I_QTY
                                                , W_ID
                                                , I_SUPPLY_W_ID
                                                , O_ID
                                                , D_ID
                                                )
                                  ) AS NEW_OL_ALL

                         WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

     FROM NEW TABLE ( INSERT INTO ORDER_LINE

                   (   OL_O_ID
                     , OL_D_ID
                     , OL_W_ID
                     , OL_NUMBER
                     , OL_I_ID
                     , OL_SUPPLY_W_ID
                     , OL_DELIVERY_D
                     , OL_QUANTITY
                     , OL_AMOUNT
                     , OL_DIST_INFO
                   )

                   INCLUDE (   I_PRICE    DECIMAL(5,2)
                             , I_NAME     CHAR(24)
                             , I_DATA     VARCHAR(50)
                             , S_DATA     VARCHAR(50)
                             , S_QUANTITY SMALLINT )

                          SELECT   O_ID
                                 , D_ID
                                 , W_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_SUPPLY_W_ID
                                 , OL_DELIVERY_D
                                 , I_QTY
                                 , TOTAL_PRICE
                                 , OL_DIST_INFO
                                 , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                          FROM DATA

                    ) AS INS
  ;

  EXEC SQL DECLARE ISOL_Local_1 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , W_ID AS I_SUPPLY_W_ID
                  , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                  , I_QTY
                  , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                     , :w_id AS W_ID
                                     , :d_id as D_ID
                                     , OL_NUMBER
                                     , I_ID
                                     , I_QTY

                               FROM  Table( VALUES

                                          ( SMALLINT( 1 ) , :id0  , :ol_quantity0 )

                                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         )
```

```
                    ) AS ITEMLIST

                  , TABLE( NEW_OL_LOCAL(   I_ID
                                         , I_QTY
                                         , W_ID
                                         , O_ID
                                         , D_ID
                                         )
                         ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
          )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    DECIMAL(5,2)
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )

                  SELECT   O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , OL_DELIVERY_D
                         , I_QTY
                         , TOTAL_PRICE
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM DATA

              ) AS INS
;

EXEC SQL DECLARE ISOL_Local_2 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                                , :w_id AS W_ID
                                , :d_id as D_ID
                                , OL_NUMBER
                                , I_ID
                                , I_QTY

                           FROM  Table( VALUES

                                   (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                 , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )

                                 ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                       ) AS ITEMLIST

                     , TABLE( NEW_OL_LOCAL(   I_ID
                                            , I_QTY
                                            , W_ID
                                            , O_ID
                                            , D_ID
                                            )
                            ) AS NEW_OL_LOCAL

                     WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
          )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    DECIMAL(5,2)
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )

                  SELECT   O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , OL_DELIVERY_D
                         , I_QTY
                         , TOTAL_PRICE
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM DATA

              ) AS INS
;

EXEC SQL DECLARE ISOL_Local_3 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                                , :w_id AS W_ID
                                , :d_id as D_ID
                                , OL_NUMBER
                                , I_ID
                                , I_QTY

                           FROM  Table( VALUES

                                   (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                 , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                 , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )

                                 ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                       ) AS ITEMLIST

                     , TABLE( NEW_OL_LOCAL(   I_ID
                                            , I_QTY
                                            , W_ID
                                            , O_ID
                                            , D_ID
                                            )
                            ) AS NEW_OL_LOCAL

                     WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
          )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    DECIMAL(5,2)
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )

                  SELECT   O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , OL_DELIVERY_D
                         , I_QTY
                         , TOTAL_PRICE
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM DATA
```

```
                ) AS INS
;

EXEC SQL DECLARE ISOL_Local_4 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                         FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                       ) AS ITEMLIST

                     , TABLE( NEW_OL_LOCAL(   I_ID
                                            , I_QTY
                                            , W_ID
                                            , O_ID
                                            , D_ID
                                            )
                       ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
          )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    DECIMAL(5,2)
                      , I_NAME     CHAR(24)
                      , I_DATA     VARCHAR(50)
                      , S_DATA     VARCHAR(50)
                      , S_QUANTITY SMALLINT )

                    SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , I_SUPPLY_W_ID
                          , OL_DELIVERY_D
                          , I_QTY
                          , TOTAL_PRICE
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                      FROM DATA

                ) AS INS
;

EXEC SQL DECLARE ISOL_Local_5 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                         FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                       ) AS ITEMLIST

                     , TABLE( NEW_OL_LOCAL(   I_ID
                                            , I_QTY
                                            , W_ID
                                            , O_ID
                                            , D_ID
                                            )
                       ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
          )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    DECIMAL(5,2)
                      , I_NAME     CHAR(24)
                      , I_DATA     VARCHAR(50)
                      , S_DATA     VARCHAR(50)
                      , S_QUANTITY SMALLINT )

                    SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , I_SUPPLY_W_ID
                          , OL_DELIVERY_D
                          , I_QTY
                          , TOTAL_PRICE
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                      FROM DATA

                ) AS INS
;

EXEC SQL DECLARE ISOL_Local_6 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                         FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                       ) AS ITEMLIST

                     , TABLE( NEW_OL_LOCAL(   I_ID
                                            , I_QTY
                                            , W_ID
                                            , O_ID
                                            , D_ID
                                            )
                       ) AS NEW_OL_LOCAL
```

```
                    WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
            )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

      FROM NEW TABLE ( INSERT INTO ORDER_LINE

                  (   OL_O_ID
                    , OL_D_ID
                    , OL_W_ID
                    , OL_NUMBER
                    , OL_I_ID
                    , OL_SUPPLY_W_ID
                    , OL_DELIVERY_D
                    , OL_QUANTITY
                    , OL_AMOUNT
                    , OL_DIST_INFO
                  )

                  INCLUDE (   I_PRICE    DECIMAL(5,2)
                            , I_NAME     CHAR(24)
                            , I_DATA     VARCHAR(50)
                            , S_DATA     VARCHAR(50)
                            , S_QUANTITY SMALLINT )

                        SELECT   O_ID
                              , D_ID
                              , W_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , OL_DELIVERY_D
                              , I_QTY
                              , TOTAL_PRICE
                              , OL_DIST_INFO
                              , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM DATA

                  ) AS INS
;

  EXEC SQL DECLARE ISOL_Local_7 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , W_ID AS I_SUPPLY_W_ID
                  , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                  , I_QTY
                  , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_QTY

                              FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                        ) AS ITEMLIST

                      , TABLE( NEW_OL_LOCAL(   I_ID
                                             , I_QTY
                                             , W_ID
                                             , O_ID
                                             , D_ID
                                             )
                        ) AS NEW_OL_LOCAL

                    WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
            )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

      FROM NEW TABLE ( INSERT INTO ORDER_LINE

                  (   OL_O_ID
                    , OL_D_ID
                    , OL_W_ID
                    , OL_NUMBER
                    , OL_I_ID
                    , OL_SUPPLY_W_ID
                    , OL_DELIVERY_D
                    , OL_QUANTITY
                    , OL_AMOUNT
                    , OL_DIST_INFO
                  )

                  INCLUDE (   I_PRICE    DECIMAL(5,2)
                            , I_NAME     CHAR(24)
                            , I_DATA     VARCHAR(50)
                            , S_DATA     VARCHAR(50)
                            , S_QUANTITY SMALLINT )

                        SELECT   O_ID
                              , D_ID
                              , W_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , OL_DELIVERY_D
                              , I_QTY
                              , TOTAL_PRICE
                              , OL_DIST_INFO
                              , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM DATA

                  ) AS INS
;

  EXEC SQL DECLARE ISOL_Local_8 CURSOR FOR

    WITH DATA AS (  SELECT  O_ID
                  , D_ID
                  , W_ID
                  , OL_NUMBER
                  , I_ID
                  , W_ID AS I_SUPPLY_W_ID
                  , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                  , I_QTY
                  , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                  , OL_DIST_INFO
                  , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT   :next_o_id as O_ID
                                    , :w_id AS W_ID
                                    , :d_id as D_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_QTY

                              FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )
                                  , (  SMALLINT( 8 )  , :id7  , :ol_quantity7  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                        ) AS ITEMLIST

                      , TABLE( NEW_OL_LOCAL(   I_ID
                                             , I_QTY
                                             , W_ID
                                             , O_ID
                                             , D_ID
                                             )
                        ) AS NEW_OL_LOCAL

                    WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
            )

    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

      FROM NEW TABLE ( INSERT INTO ORDER_LINE

                  (   OL_O_ID
                    , OL_D_ID
                    , OL_W_ID
                    , OL_NUMBER
                    , OL_I_ID
                    , OL_SUPPLY_W_ID
                    , OL_DELIVERY_D
                    , OL_QUANTITY
                    , OL_AMOUNT
                    , OL_DIST_INFO
                  )

                  INCLUDE (   I_PRICE    DECIMAL(5,2)
                            , I_NAME     CHAR(24)
                            , I_DATA     VARCHAR(50)
                            , S_DATA     VARCHAR(50)
                            , S_QUANTITY SMALLINT )

                        SELECT   O_ID
                              , D_ID
                              , W_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , OL_DELIVERY_D
                              , I_QTY
                              , TOTAL_PRICE
                              , OL_DIST_INFO
                              , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM DATA
```

```
            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_9 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                        FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )
                                  , (  SMALLINT( 8 )  , :id7  , :ol_quantity7  )
                                  , (  SMALLINT( 9 )  , :id8  , :ol_quantity8  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                     ) AS ITEMLIST

                   , TABLE( NEW_OL_LOCAL(   I_ID
                                          , I_QTY
                                          , W_ID
                                          , O_ID
                                          , D_ID
                                          )
                          ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
          )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    DECIMAL(5,2)
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )

                       SELECT   O_ID
                            , D_ID
                            , W_ID
                            , OL_NUMBER
                            , I_ID
                            , I_SUPPLY_W_ID
                            , OL_DELIVERY_D
                            , I_QTY
                            , TOTAL_PRICE
                            , OL_DIST_INFO
                            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                       FROM DATA

                ) AS INS
;

EXEC SQL DECLARE ISOL_Local_10 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                        FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )
                                  , (  SMALLINT( 8 )  , :id7  , :ol_quantity7  )
                                  , (  SMALLINT( 9 )  , :id8  , :ol_quantity8  )
                                  , (  SMALLINT( 10 ) , :id9  , :ol_quantity9  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                     ) AS ITEMLIST

                   , TABLE( NEW_OL_LOCAL(   I_ID
                                          , I_QTY
                                          , W_ID
                                          , O_ID
                                          , D_ID
                                          )
                          ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
          )

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    DECIMAL(5,2)
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )

                       SELECT   O_ID
                            , D_ID
                            , W_ID
                            , OL_NUMBER
                            , I_ID
                            , I_SUPPLY_W_ID
                            , OL_DELIVERY_D
                            , I_QTY
                            , TOTAL_PRICE
                            , OL_DIST_INFO
                            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                       FROM DATA

                ) AS INS
;

EXEC SQL DECLARE ISOL_Local_11 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                        FROM  Table( VALUES

                                    (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )
                                  , (  SMALLINT( 8 )  , :id7  , :ol_quantity7  )
```

```
                                , (  SMALLINT( 9 )  , :id8  , :ol_quantity8  )
                                , (  SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                                , (  SMALLINT( 11 ) , :id10 , :ol_quantity10 )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                            ) AS ITEMLIST

                          , TABLE( NEW_OL_LOCAL(   I_ID
                                                 , I_QTY
                                                 , W_ID
                                                 , O_ID
                                                 , D_ID
                                                )
                                 ) AS NEW_OL_LOCAL

                         WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

      FROM NEW TABLE ( INSERT INTO ORDER_LINE

                    (   OL_O_ID
                      , OL_D_ID
                      , OL_W_ID
                      , OL_NUMBER
                      , OL_I_ID
                      , OL_SUPPLY_W_ID
                      , OL_DELIVERY_D
                      , OL_QUANTITY
                      , OL_AMOUNT
                      , OL_DIST_INFO
                    )

                    INCLUDE (   I_PRICE    DECIMAL(5,2)
                              , I_NAME     CHAR(24)
                              , I_DATA     VARCHAR(50)
                              , S_DATA     VARCHAR(50)
                              , S_QUANTITY SMALLINT )

                         SELECT   O_ID
                                , D_ID
                                , W_ID
                                , OL_NUMBER
                                , I_ID
                                , I_SUPPLY_W_ID
                                , OL_DELIVERY_D
                                , I_QTY
                                , TOTAL_PRICE
                                , OL_DIST_INFO
                                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                         FROM DATA

                    ) AS INS
;

EXEC SQL DECLARE ISOL_Local_12 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , W_ID AS I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                         FROM  Table( VALUES

                                      (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                    , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                    , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                    , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                    , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                    , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                    , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )
                                    , (  SMALLINT( 8 )  , :id7  , :ol_quantity7  )
                                    , (  SMALLINT( 9 )  , :id8  , :ol_quantity8  )
                                    , (  SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                                    , (  SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                                    , (  SMALLINT( 12 ) , :id11 , :ol_quantity11 )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                            ) AS ITEMLIST

                          , TABLE( NEW_OL_LOCAL(   I_ID
                                                 , I_QTY
                                                 , W_ID
                                                 , O_ID
                                                 , D_ID
                                                )
                                 ) AS NEW_OL_LOCAL

                         WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

      FROM NEW TABLE ( INSERT INTO ORDER_LINE

                    (   OL_O_ID
                      , OL_D_ID
                      , OL_W_ID
                      , OL_NUMBER
                      , OL_I_ID
                      , OL_SUPPLY_W_ID
                      , OL_DELIVERY_D
                      , OL_QUANTITY
                      , OL_AMOUNT
                      , OL_DIST_INFO
                    )

                    INCLUDE (   I_PRICE    DECIMAL(5,2)
                              , I_NAME     CHAR(24)
                              , I_DATA     VARCHAR(50)
                              , S_DATA     VARCHAR(50)
                              , S_QUANTITY SMALLINT )

                         SELECT   O_ID
                                , D_ID
                                , W_ID
                                , OL_NUMBER
                                , I_ID
                                , I_SUPPLY_W_ID
                                , OL_DELIVERY_D
                                , I_QTY
                                , TOTAL_PRICE
                                , OL_DIST_INFO
                                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                         FROM DATA

                    ) AS INS
;

EXEC SQL DECLARE ISOL_Local_13 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
                , D_ID
                , W_ID
                , OL_NUMBER
                , I_ID
                , W_ID AS I_SUPPLY_W_ID
                , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
                , I_QTY
                , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                         FROM  Table( VALUES

                                      (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                    , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                    , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                    , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                    , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                    , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                    , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )
                                    , (  SMALLINT( 8 )  , :id7  , :ol_quantity7  )
                                    , (  SMALLINT( 9 )  , :id8  , :ol_quantity8  )
                                    , (  SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                                    , (  SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                                    , (  SMALLINT( 12 ) , :id11 , :ol_quantity11 )
                                    , (  SMALLINT( 13 ) , :id12 , :ol_quantity12 )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                            ) AS ITEMLIST

                          , TABLE( NEW_OL_LOCAL(   I_ID
                                                 , I_QTY
                                                 , W_ID
                                                 , O_ID
                                                 , D_ID
                                                )
                                 ) AS NEW_OL_LOCAL

                         WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
                )


    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

      FROM NEW TABLE ( INSERT INTO ORDER_LINE

                    (   OL_O_ID
                      , OL_D_ID
                      , OL_W_ID
                      , OL_NUMBER
                      , OL_I_ID
                      , OL_SUPPLY_W_ID
```

```
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
               )

               INCLUDE (   I_PRICE    DECIMAL(5,2)
                       , I_NAME     CHAR(24)
                       , I_DATA     VARCHAR(50)
                       , S_DATA     VARCHAR(50)
                       , S_QUANTITY SMALLINT )

                     SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , I_SUPPLY_W_ID
                          , OL_DELIVERY_D
                          , I_QTY
                          , TOTAL_PRICE
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                     FROM DATA

             ) AS INS
;

EXEC SQL DECLARE ISOL_Local_14 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                           , :w_id AS W_ID
                           , :d_id as D_ID
                           , OL_NUMBER
                           , I_ID
                           , I_QTY

                       FROM  Table( VALUES

                                ( SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                               , ( SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                               , ( SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                               , ( SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                               , ( SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                               , ( SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                               , ( SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                               , ( SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                               , ( SMALLINT( 9 ) , :id8  , :ol_quantity8  )
                               , ( SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                               , ( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                               , ( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
                               , ( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
                               , ( SMALLINT( 14 ) , :id13 , :ol_quantity13 )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                       ) AS ITEMLIST

                     , TABLE( NEW_OL_LOCAL(   I_ID
                                    , I_QTY
                                    , W_ID
                                    , O_ID
                                    , D_ID
                                   )
                            ) AS NEW_OL_LOCAL

                     WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
               )


  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

   FROM NEW TABLE ( INSERT INTO ORDER_LINE

               (   OL_O_ID
                 , OL_D_ID
                 , OL_W_ID
                 , OL_NUMBER
                 , OL_I_ID
                 , OL_SUPPLY_W_ID
                 , OL_DELIVERY_D
                 , OL_QUANTITY
                 , OL_AMOUNT
                 , OL_DIST_INFO
               )

               INCLUDE (   I_PRICE    DECIMAL(5,2)
                       , I_NAME     CHAR(24)
                       , I_DATA     VARCHAR(50)
                       , S_DATA     VARCHAR(50)
                       , S_QUANTITY SMALLINT )

                     SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , I_SUPPLY_W_ID
                          , OL_DELIVERY_D
                          , I_QTY
                          , TOTAL_PRICE
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                     FROM DATA

             ) AS INS
;

EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

  WITH DATA AS (  SELECT  O_ID
            , D_ID
            , W_ID
            , OL_NUMBER
            , I_ID
            , W_ID AS I_SUPPLY_W_ID
            , (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
            , I_QTY
            , ( I_PRICE * I_QTY ) AS TOTAL_PRICE
            , OL_DIST_INFO
            , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                  FROM ( SELECT   :next_o_id as O_ID
                           , :w_id AS W_ID
                           , :d_id as D_ID
                           , OL_NUMBER
                           , I_ID
                           , I_QTY

                       FROM  Table( VALUES

                                ( SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                               , ( SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                               , ( SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                               , ( SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                               , ( SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                               , ( SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                               , ( SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                               , ( SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                               , ( SMALLINT( 9 ) , :id8  , :ol_quantity8  )
                               , ( SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                               , ( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                               , ( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
                               , ( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
                               , ( SMALLINT( 14 ) , :id13 , :ol_quantity13 )
                               , ( SMALLINT( 15 ) , :id14 , :ol_quantity14 )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                       ) AS ITEMLIST

                     , TABLE( NEW_OL_LOCAL(   I_ID
                                    , I_QTY
                                    , W_ID
                                    , O_ID
                                    , D_ID
                                   )
                            ) AS NEW_OL_LOCAL

                     WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
               )


SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

  FROM NEW TABLE ( INSERT INTO ORDER_LINE

               (   OL_O_ID
                 , OL_D_ID
                 , OL_W_ID
                 , OL_NUMBER
                 , OL_I_ID
                 , OL_SUPPLY_W_ID
                 , OL_DELIVERY_D
                 , OL_QUANTITY
                 , OL_AMOUNT
                 , OL_DIST_INFO
               )

               INCLUDE (   I_PRICE    DECIMAL(5,2)
                       , I_NAME     CHAR(24)
                       , I_DATA     VARCHAR(50)
                       , S_DATA     VARCHAR(50)
                       , S_QUANTITY SMALLINT )

                     SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , I_SUPPLY_W_ID
                          , OL_DELIVERY_D
                          , I_QTY
                          , TOTAL_PRICE
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                     FROM DATA
```

```
                    ) AS INS
  ;

  // Start processing

  in_neword = (struct in_neword_struct *)  pin ;
  neword    = (struct out_neword_struct *) pout ;

#ifdef DEBUGIT
  new_debug( neword, in_neword, "SP upon entry");
#endif

  // Using I_PRICE == 0 as a flag to the client that the ITEM was not fetched (hence bad).

  for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < in_neword->s_O_OL_CNT ;
inputItemArrayIndex++ )
  {
    i_priceArray[ inputItemArrayIndex ] = 0 ;
  }

  neword->deadlocks = -1 ;

 retry_tran:

  neword->deadlocks++ ;

  EXEC SQL

    SELECT D_TAX, D_NEXT_O_ID  INTO :dist_tax , :next_o_id

      FROM OLD TABLE ( UPDATE DISTRICT

                          SET  D_NEXT_O_ID = D_NEXT_O_ID + 1

                        WHERE D_W_ID = :w_id
                          AND D_ID = :d_id
                ) AS OT
  ;

  if ( sqlca.sqlcode != 0 )
  {
    DLCHK( retry_tran );
    sqlerror( NEWORD_SQL, "DISTRICT", __FILE__, __LINE__, &sqlca ) ;
    goto ferror;
  }

  #define NEW_CURSOR_OPEN_ERROR                                          \
  {                                                                      \
    if( sqlca.sqlcode != 0    )                                          \
    {                                                                    \
      goto sql_error ;                                                   \
    }                                                                    \
  }

  #define NEW_CURSOR_ERROR                                               \
  {                                                                      \
    if( sqlca.sqlcode == 0    )                                          \
    {                                                                    \
      neword->s_O_OL_CNT ++ ;                                            \
    }                                                                    \
    else                                                                 \
    if( sqlca.sqlcode == +100 )                                          \
    {                                                                    \
      break ;                                                            \
    }                                                                    \
    else                                                                 \
      goto sql_error ;                                                   \
  }
```

```
  if ( allLocal )
  {
    switch( inputItemCount )
    {
      case 1:
        EXEC SQL OPEN ISOL_Local_1 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_1
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 2:
        EXEC SQL OPEN ISOL_Local_2 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_2
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 3:
        EXEC SQL OPEN ISOL_Local_3 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_3
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 4:
        EXEC SQL OPEN ISOL_Local_4 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_4
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 5:
        EXEC SQL OPEN ISOL_Local_5 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_5
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 6:
        EXEC SQL OPEN ISOL_Local_6 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 7:
        EXEC SQL OPEN ISOL_Local_7 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 8:
        EXEC SQL OPEN ISOL_Local_8 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 9:
        EXEC SQL OPEN ISOL_Local_9 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 10:
        EXEC SQL OPEN ISOL_Local_10 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 11:
        EXEC SQL OPEN ISOL_Local_11 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 12:
        EXEC SQL OPEN ISOL_Local_12 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 13:
        EXEC SQL OPEN ISOL_Local_13 ;
        NEW_CURSOR_OPEN_ERROR
```

```
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Local_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 14:
          EXEC SQL OPEN ISOL_Local_14 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Local_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 15:
          EXEC SQL OPEN ISOL_Local_15 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Local_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;

        default:
          sqlerror(NEWORD_SQL, "Default switch on local  orderline/stock/index", __FILE__,
__LINE__, &sqlca );
          goto ferror;
      }
    }
    else
    {
      switch( inputItemCount )
      {
        case 1:
          EXEC SQL OPEN ISOL_Remote_1 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_1
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 2:
          EXEC SQL OPEN ISOL_Remote_2 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_2
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 3:
          EXEC SQL OPEN ISOL_Remote_3 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_3
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 4:
          EXEC SQL OPEN ISOL_Remote_4 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_4
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 5:
          EXEC SQL OPEN ISOL_Remote_5 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_5
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 6:
          EXEC SQL OPEN ISOL_Remote_6 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 7:
          EXEC SQL OPEN ISOL_Remote_7 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 8:
          EXEC SQL OPEN ISOL_Remote_8 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 9:
          EXEC SQL OPEN ISOL_Remote_9 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 10:
          EXEC SQL OPEN ISOL_Remote_10 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 11:
          EXEC SQL OPEN ISOL_Remote_11 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 12:
          EXEC SQL OPEN ISOL_Remote_12 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 13:
          EXEC SQL OPEN ISOL_Remote_13 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 14:
          EXEC SQL OPEN ISOL_Remote_14 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 15:
          EXEC SQL OPEN ISOL_Remote_15 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ;
inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;

        default:
```

```
        sqlerror(NEWORD_SQL, "Default switch on remote orderline/stock/index",
__FILE__, __LINE__, &sqlca);
        goto ferror;
      }
    }

    for ( inputItemArrayIndex = 0 ;
          inputItemArrayIndex < in_neword->s_O_OL_CNT  // from input
        && i_priceArray[ inputItemArrayIndex ] != 0 ;
          inputItemArrayIndex++ )
    {
      // s_I_NAME, and s_S_QUANTITY already set as output host variables

      neword->item[ inputItemArrayIndex ].s_I_PRICE = i_priceArray[ inputItemArrayIndex ] ;

      if (    is_ORIGINAL( s_dataArray[ inputItemArrayIndex ].data,
s_dataArray[ inputItemArrayIndex ].len )
           && is_ORIGINAL( i_dataArray[ inputItemArrayIndex ].data,
i_dataArray[ inputItemArrayIndex ].len )    )
      {
        neword->item[ inputItemArrayIndex ].s_brand_generic = 'B';
      }
      else
      {
        neword->item[ inputItemArrayIndex ].s_brand_generic = 'G';
      }
    }

    EXEC SQL

      SELECT W_TAX, C_DISCOUNT, C_LAST, C_CREDIT, O_ENTRY_D

        INTO :ware_tax, :c_discount, :c_last, :c_credit, :o_entry_d

        FROM TABLE ( NEW_WH (   :next_o_id
                              , :w_id
                              , :d_id
                              , :c_id
                              , :inputItemCount
                              , :allLocal
                            )
                   ) AS NEW_WH_TABLE
    ;

    if ( sqlca.sqlcode == 0 )
    {
      if ( neword->s_O_OL_CNT == in_neword->s_O_OL_CNT  )
      {
        neword->s_transtatus = TRAN_OK ;

        EXEC SQL COMMIT;

        if( sqlca.sqlcode != 0 )
        {
          sqlerror(NEWORD_SQL, "COMMIT", __FILE__, __LINE__ , &sqlca ) ;
          goto ferror;
        }
      }
      else
      {
        neword->s_transtatus = INVALID_ITEM ;

        EXEC SQL ROLLBACK WORK ;

        if ( sqlca.sqlcode != 0 )
        {
          neword->s_transtatus = FATAL_SQLERROR;

          sqlerror(NEWORD_SQL, "ROLLBACK FAILED (INVALID ITEM)", __FILE__,
__LINE__ , &sqlca);
          // no point in ferror
        }
      }
    }
    else
    {
      DLCHK( retry_tran );

      sqlerror( NEWORD_SQL, "NEW_WH", __FILE__, __LINE__ , &sqlca);
      goto ferror;
    }

    /*---------------------------------------------*/
    /*  Return to client                           */
    /*---------------------------------------------*/

mexit:

    if ( sqlca.sqlcode >= 0 )
    {
      storedProcRc = SQLZ_HOLD_PROC ;
    }
    else
    {
      storedProcRc = SQLZ_DISCONNECT_PROC ;
    }

#ifdef DEBUGIT
    new_debug( neword, in_neword, "SP prior to return");
#endif

    return ( storedProcRc ) ;

sql_error:

    {
      char tempstr[ 4096 ] ;

      DLCHK( retry_tran ) ;

      sprintf( tempstr, "inputItemCount=%d, :next_o_id=%d, :d_id=%d, :w_id=%d",
inputItemCount, next_o_id, d_id, w_id ) ;
      sqlerror( NEWORD_SQL, tempstr , __FILE__, __LINE__, &sqlca ) ;
    }

ferror:

    neword->s_transtatus = FATAL_SQLERROR;

    EXEC SQL ROLLBACK WORK;

    if ( sqlca.sqlcode != 0 )
    {
      sqlerror( NEWORD_SQL, "ROLLBACK FAILED", __FILE__, __LINE__ , &sqlca ) ;
    }


    goto mexit ;
}

/*
** A little function to search for the string "ORIGINAL" given a string and
** it's length
*/
static unsigned char  skip[256] = {8,8,8,8,8,8,8,8,8,8,  /*0-9*/
                                   8,8,8,8,8,8,8,8,8,8,  /*10-19*/
                                   8,8,8,8,8,8,8,8,8,8,  /*20-29*/
                                   8,8,8,8,8,8,8,8,8,8,  /*30-39*/
                                   8,8,8,8,8,8,8,8,8,8,  /*40-49*/
                                   8,8,8,8,8,8,8,8,8,8,  /*50-59*/
                                   8,8,8,8,8,1,8,8,8,8,  /*60-69*/
                                   8,4,8,3,8,8,0,8,2,7,  /*70-79*/
                                   8,8,6,8,8,8,8,8,8,8,  /*80-89*/
                                   8,8,8,8,8,8,8,8,8,8,  /*90-99*/
                                   8,8,8,8,8,8,8,8,8,8,  /*100-109*/
                                   8,8,8,8,8,8,8,8,8,8,  /*110-119*/
                                   8,8,8,8,8,8,8,8,8,8,  /*120-129*/
                                   8,8,8,8,8,8,8,8,8,8,  /*130-139*/
                                   8,8,8,8,8,8,8,8,8,8,  /*140-149*/
                                   8,8,8,8,8,8,8,8,8,8,  /*150-159*/
                                   8,8,8,8,8,8,8,8,8,8,  /*160-169*/
                                   8,8,8,8,8,8,8,8,8,8,  /*170-179*/
                                   8,8,8,8,8,8,8,8,8,8,  /*180-189*/
                                   8,8,8,8,8,8,8,8,8,8,  /*190-199*/
                                   8,8,8,8,8,8,8,8,8,8,  /*200-209*/
                                   8,8,8,8,8,8,8,8,8,8,  /*210-219*/
                                   8,8,8,8,8,8,8,8,8,8,  /*220-229*/
                                   8,8,8,8,8,8,8,8,8,8,  /*230-239*/
                                   8,8,8,8,8,8,8,8,8,8,  /*240-249*/
                                   8,8,8,8,8};           /*250-254*/

static int is_ORIGINAL( char *string, short length )
{
  char           *cur_string;
  char           *end_string;
  unsigned char   *skips;
  int             skip_dist;
  int             result = 0;

  cur_string = string+7;
  end_string = string + length;
  skips = skip;

  while (cur_string < end_string)
   {
     skip_dist = skips[*cur_string];
     while ( (skip_dist > 0) && (cur_string < end_string) )
       {
         skip_dist = skips[*(cur_string += skip_dist)];
       }

     if (cur_string >= end_string)
       goto exit;

     if ( cur_string[-4] != 'G' )
       goto noMatch;

     if ( memcmp( cur_string-7, "ORIGINAL", 8 ) == 0 )
       {
         result = 1;
         goto exit;
       }
noMatch:
     cur_string += 8;
   } /* end while */

exit:
  return ( result ) ;
}

// ---------------------------------------------------------------
// Order Status SERVER
// ---------------------------------------------------------------

#undef w_id
#undef d_id
#undef c_id_input
#undef o_id
#undef o_entry_d
#undef o_carrier_d
```

```
#undef c_id
#undef c_first
#undef c_middle
#undef c_last
#undef c_balance

SQL_API_RC order_status_internal( char *pin, char *pout )
{
   struct in_ordstat_struct  * in_ordstat = (struct in_ordstat_struct *) pin ;
   struct out_ordstat_struct * ordstat = (struct out_ordstat_struct *) pout ;

   struct sqlca sqlca ;

   EXEC SQL BEGIN DECLARE SECTION;

    // From input values

    //##sqlint32   w_id ;
    //##short      d_id;
    sqlint32   c_id_input ;

    struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;

    // From queries

    // From initial query

    sqlint32   o_id ;
    //##sqlint32   c_id ;
    short      o_carrier_id ;
    //##sqlint64   o_entry_d ;

    char   c_first[ 16 ] ;
    char   c_middle[ 2 ] ;
    //##char   c_last[ 16 ] ;
    double c_balance ;

    // From cursor

    sqlint32   ol_i_id ;
    sqlint32   ol_supply_w_id ;
    short      ol_quantity ;
    float   ol_amount ;
    char    ol_delivery_d [27] ;
    //##char    o_entry_d[ 27 ] ;

   EXEC SQL END DECLARE SECTION;

   //##struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;

   int     storedProcRc ;
   int     itemArrayIndex = 0 ;

   #define w_id                in_ordstat->s_W_ID ;
   #define d_id                in_ordstat->s_D_ID ;
   #define c_id_input          in_ordstat->s_C_ID
   #define o_id                ordstat->s_O_ID
   #define o_entry_d           ordstat->s_O_ENTRY_D_time
   #define o_carrier_id        ordstat->s_O_CARRIER_ID
   #define c_id                ordstat->s_C_ID
   #define c_first             ordstat->s_C_FIRST
   #define c_middle            ordstat->s_C_MIDDLE
   #define c_last              ordstat->s_C_LAST
   #define c_balance           ordstat->s_C_BALANCE

   EXEC SQL DECLARE read_orderline_cur CURSOR FOR

      SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY, OL_AMOUNT,
OL_DELIVERY_D
        FROM ORDER_LINE

        WHERE OL_W_ID = :w_id
          AND OL_D_ID = :d_id
          AND OL_O_ID = :o_id

        FOR FETCH ONLY ;

   ordstat->deadlocks = -1 ;

#ifdef DEBUGIT
   ord_debug(ordstat, in_ordstat, "SP upon entry");
#endif

retry_tran:

   ordstat->deadlocks ++ ;

   if ( c_id_input == 0 )
   {
      c_last_input.len = strlen( in_ordstat->s_C_LAST ) ;
      memcpy( c_last_input.data , in_ordstat->s_C_LAST , c_last_input.len ) ;

      EXEC SQL

         SELECT O_ID, O_CARRIER_ID, O_ENTRY_D, C_BALANCE, C_FIRST, C_MIDDLE,
C_ID

           INTO :o_id, :o_carrier_id , :o_entry_d , :c_balance, :c_first, :c_middle, :c_id

           FROM TABLE ( ORD_C_LAST(   :w_id
                                    , :d_id
                                    , :c_last_input
                                  )
                      ) AS ORD_C_LAST
      ;
   }
   else
   {
      EXEC SQL

         SELECT O_ID, O_CARRIER_ID, O_ENTRY_D , C_BALANCE, C_FIRST,
C_MIDDLE ,C_LAST

           INTO :o_id, :o_carrier_id , :o_entry_d , :c_balance, :c_first, :c_middle, :c_last

           FROM TABLE ( ORD_C_ID(   :w_id
                                  , :d_id
                                  , :c_id_input
                                )
                      ) AS ORD_C_ID
      ;
   }

   if ( sqlca.sqlcode != 0 )
   {
      DLCHK( retry_tran );
      sqlerror( ORDSTAT_SQL, "READ CUST and ORDERS", __FILE__, __LINE__ ,
&sqlca ) ;
      goto ferror;
   }

   /*---------------------------------------------*/
   /*  Read ORDER_LINEs                           */
   /*---------------------------------------------*/

   EXEC SQL OPEN read_orderline_cur ;

   if ( sqlca.sqlcode != 0 )
   {
      DLCHK( retry_tran );
      sqlerror(ORDSTAT_SQL, "OPEN CURSOR read_orderline_cur", __FILE__, __LINE__
&sqlca ) ;
      goto ferror;
   }

   itemArrayIndex = 0 ;
   {
      do
      {
         EXEC SQL FETCH read_orderline_cur

                  INTO :ol_i_id , :ol_supply_w_id , :ol_quantity , :ol_amount , :ol_delivery_d ;

         if ( sqlca.sqlcode == 0 )
         {
            ordstat->item[ itemArrayIndex ].s_OL_I_ID           = ol_i_id ;
            ordstat->item[ itemArrayIndex ].s_OL_SUPPLY_W_ID    = ol_supply_w_id ;
            ordstat->item[ itemArrayIndex ].s_OL_QUANTITY       = ol_quantity ;
            ordstat->item[ itemArrayIndex ].s_OL_AMOUNT         = ol_amount ;
            strcpy(ordstat->item[ itemArrayIndex ].s_OL_DELIVERY_D_time, ol_delivery_d) ;

            itemArrayIndex++;
         }
         else
         if (sqlca.sqlcode < 0 )
         {
            DLCHK( retry_tran ) ;
            sqlerror( ORDSTAT_SQL, "FETCH CURSOR read_orderline_cur" , __FILE__,
__LINE__ , &sqlca ) ;
            goto ferror ;
         }
      }
      while ( sqlca.sqlcode == 0 ) ;
   }

   ordstat->s_ol_cnt = itemArrayIndex ;

   EXEC SQL COMMIT ;

   if ( sqlca.sqlcode == 0 )
   {
      ordstat->s_transtatus = TRAN_OK ;
   }
   else
   {
      DLCHK( retry_tran );
      sqlerror(ORDSTAT_SQL, "COMMIT", __FILE__, __LINE__ , &sqlca);
      goto ferror ;
   }

mexit:

   if ( sqlca.sqlcode >= 0 )
   {
      storedProcRc = SQLZ_HOLD_PROC ;
   }
   else
   {
      storedProcRc = SQLZ_DISCONNECT_PROC ;
   }

#ifdef DEBUGIT
   ord_debug(ordstat, in_ordstat, "SP prior to return");
#endif

   return ( storedProcRc ) ;

ferror:
```

```
   ordstat->s_transtatus = FATAL_SQLERROR ;

   EXEC SQL ROLLBACK WORK ;

   if ( sqlca.sqlcode != 0 )
   {
     sqlerror(ORDSTAT_SQL, "ROLLBACK FAILED", __FILE__, __LINE__, &sqlca);
   }

   goto mexit;
}

// --------------------------------------------------------------------
// Delivery SERVER
// --------------------------------------------------------------------

#undef d_id
#undef c_id
#undef w_id
#undef o_carrier_id
#undef ol_delivery_d

SQL_API_RC delivery_internal ( char * pin, char * pout )
{
   struct in_delivery_struct * in_delivery = (struct in_delivery_struct *)  pin ;
   struct out_delivery_struct * delivery   = (struct out_delivery_struct *) pout ;

   struct sqlca sqlca ;

   int storedProcRc ;

   short district_id ;
   sqlint32  customer_id ;

   EXEC SQL BEGIN DECLARE SECTION;

     // input

     //##sqlint32   w_id ;
     //##short      d_id ;
     //##sqlint32   c_id ;
     //##short      o_carrier_id ;
     //##sqlint64   ol_delivery_d ;

     // output

     short      no_o_id_indicator = 0 ;
     sqlint32   no_o_id ;

   EXEC SQL END DECLARE SECTION;

   #define d_id          district_id
   #define c_id          customer_id

   #define w_id          in_delivery->s_W_ID
   #define o_carrier_id  in_delivery->s_O_CARRIER_ID
   #define ol_delivery_d in_delivery->s_O_DELIVERY_D_time

   delivery->deadlocks = -1 ;

#ifdef DEBUGIT
   del_debug( delivery, in_delivery, "SP upon entry");
#endif

   // Deadlock Handling
   // -----------------
   // Since we COMMIT inside the for() loop, we must take special
   // care while handling deadlocks.  This is best explained by
   // an example.
   //
   // Assume we deadlock on d_id=6.  This means that an order from the
   // first 5 districts have already been delivered.  We will then
   // restart the loop (retry_tran).  However, the loop will restart
   // at d_id = 1!  This means that the second (and all subsequent)
   // time through the loop, we will deliver orders for districts that
   // have already been delivered, with the net result being more than
   // 10 orders being delivered.
   //
   // The solution to this problem is to initialize the starting point
   // of the loop *before* the retry_tran label.  This will ensure that
   // if we deadlock, we will restart the loop with the same district
   // that we deadlocked on, and we won't deliver any extra orders.
   //
   // NOTE: If we ever change this back to one COMMIT per transaction
   // (instead of one COMMIT per iteration), then the initialization
   // of d_id must be moved back into the for loop.  (A rollback due
   // to deadlock in this case would rollback all delivered orders so
   // far, so we'd need to re-deliver them all on the next iteration.)

   d_id = 1;

retry_tran:

   delivery->deadlocks++;

   for ( ;  d_id <= DISTRICTS_PER_WAREHOUSE ; d_id++ )
   {
     no_o_id = 0 ;
     no_o_id_indicator = 0 ;

       EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

         SELECT O_ID

        INTO :no_o_id :no_o_id_indicator

          FROM TABLE (  DEL( :w_id , :d_id , :o_carrier_id ) ) AS T ;

         COMMIT ;

       END COMPOUND ;

     if ( sqlca.sqlcode == 0 )
     {
       delivery->s_O_ID[ d_id - 1 ] = no_o_id ;
     }
     else
     {
       DLCHK( retry_tran );

       sqlerror( DELIVERY_SQL , "DELIVERY", __FILE__, __LINE__ , &sqlca);
       goto ferror ;
     }
   }

   delivery->s_transtatus = TRAN_OK ;

mexit:

   if ( sqlca.sqlcode >= 0 )
   {
     storedProcRc = SQLZ_HOLD_PROC ;
   }
   else
   {
     storedProcRc = SQLZ_DISCONNECT_PROC ;
   }

#ifdef DEBUGIT
   del_debug( delivery, in_delivery, "SP prior to return");
#endif

   return ( storedProcRc ) ;

ferror:

   delivery->s_transtatus = FATAL_SQLERROR ;

   EXEC SQL ROLLBACK WORK ;

   if ( sqlca.sqlcode != 0 )
   {
     sqlerror( DELIVERY_SQL, "ROLLBACK FAILED", __FILE__, __LINE__ , &sqlca ) ;
   }


   goto mexit ;
}

// --------------------------------------------------------------------
// Stored Procedure Stubs
// --------------------------------------------------------------------

SQL_API_RC SQL_API_FN news( char *pin, char *pout )
{
   return new_order_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN ords( char *pin, char *pout )
{
   return order_status_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN dels ( char * pin, char * pout )
{
   return delivery_internal( pin, pout ) ;
}
```

Src.Srv/uncat-func.ddl

```
------------------------------------------------------------------------------
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
------------------------------------------------------------------------------
--
-- uncat-func.ddl - Drop table function DDL
--
--

-- STOCK LEVEL
DROP SPECIFIC FUNCTION STOCK_LEVEL %
-- DELIVERY
DROP SPECIFIC FUNCTION DELIVERY %
-- ORDER STATUS
DROP SPECIFIC FUNCTION ORD_C_LAST %
DROP SPECIFIC FUNCTION ORD_C_ID %
-- PAYMENT
DROP SPECIFIC FUNCTION PAY_C_LAST %
DROP SPECIFIC FUNCTION PAY_C_ID %
-- NEW ORDER
DROP SPECIFIC FUNCTION NEW_OL_ALL %
```

```
DROP SPECIFIC FUNCTION NEW_OL_LOCAL %
DROP SPECIFIC FUNCTION NEW_WH %

DROP PROCEDURE news
        (varchar(262),varchar(682));
DROP PROCEDURE news
        (varchar(270),varchar(662));
DROP PROCEDURE news;
```

Src.Srv/uncat-proc.ddl

```
DROP PROCEDURE pays;

DROP PROCEDURE ords
        (varchar(42),varchar(822));
DROP PROCEDURE ords
        (varchar(42),varchar(446));
DROP PROCEDURE ords;

DROP PROCEDURE dels
        (varchar(14),varchar(50));
DROP PROCEDURE dels
        (varchar(22),varchar(50));
DROP PROCEDURE dels;



DROP PROCEDURE stks;
```

include/db2tpcc.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * db2tpcc.h - Macros and Miscellany
 *
 */

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

/* ************************************************************************** */
/* Transaction Return Codes (s_transtatus)                                    */
/* ************************************************************************** */

#define INVALID_ITEM            100
#define TRAN_OK                 0
#define FATAL_SQLERROR          -1

/* ************************************************************************** */
/* Definition of Unused and Bad Items                                         */
/* ************************************************************************** */
/* Define unused item ID to be 0. This allows the SUT to determine the     */
/* number of items in the order as required by 2.4.1.3 and 2.4.2.2 since   */
/* the assumption that any item with OL_I_ID = 0 is unused will be true.   */
/* This in turn requires that the value used for an invalid item is        */
/* equal to ITEMS + 1.                                                     */
/* ************************************************************************** */

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****************************************************************************/
/* NURand Constants                                                          */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.               */
/* Analysis indicates that a C_LAST delta of 85 is optimal.                  */
/*****************************************************************************/
#define C_C_LAST_RUN  88
#define C_C_LAST_LOAD 173
#define C_C_ID    319
#define C_OL_I_ID   3849
#define A_C_LAST    255
#define A_C_ID    1023
#define A_OL_I_ID   8191

/*****************************************************************************/
/* Transaction Type Identifiers                                              */
/*****************************************************************************/

#define CLIENT_SQL   0
#define NEWORD_SQL   1
#define PAYMENT_SQL  2
#define ORDSTAT_SQL  3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   in_items_struct {
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad1[3];
  } in_item[15];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  s_O_OL_CNT;    /* init by SUT */
  int16_t  s_all_local;
  int16_t  duplicate_items;
};

struct out_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   items_struct {
    float  s_I_PRICE;
    float  s_OL_AMOUNT;
    int16_t  s_S_QUANTITY;
    int16_t  pad2;
    char     s_I_NAME[25];
    char     s_brand_generic;
  } item[15];
  float  s_W_TAX;
  float  s_D_TAX;
  float  s_C_DISCOUNT;
  float  s_total_amount;
  int32_t  s_O_ID;
  int16_t  s_O_OL_CNT;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_LAST[17];
  char     s_C_CREDIT[3];
  char     s_O_ENTRY_D_time[27];

};

struct in_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  float  s_H_AMOUNT;
  int32_t  s_W_ID;
  int32_t  s_C_W_ID;
  int32_t  s_C_ID;
  int16_t  s_C_D_ID;
  int16_t  s_D_ID;
  char     s_C_LAST[17];
};

struct out_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  double  s_C_CREDIT_LIM;
  double  s_C_BALANCE;
  float  s_C_DISCOUNT;
  int32_t  s_C_ID;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_W_STREET_1[21];
  char     s_W_STREET_2[21];
  char     s_W_CITY[21];
  char     s_W_STATE[3];
  char     s_W_ZIP[10];
  char     s_D_STREET_1[21];
  char     s_D_STREET_2[21];
  char     s_D_CITY[21];
  char     s_D_STATE[3];
  char     s_D_ZIP[10];
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_C_STREET_1[21];
  char     s_C_STREET_2[21];
  char     s_C_CITY[21];
  char     s_C_STATE[3];
  char     s_C_ZIP[10];
  char     s_C_PHONE[17];
  char     s_C_CREDIT[3];
  char     s_C_DATA[201];
  char     s_H_DATE_time[27];
  char     s_C_SINCE_time[27];
};

struct in_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  pad1[3];
  char     s_C_LAST[17];
};

struct out_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
```

```
  double  s_C_BALANCE;
  int32_t  s_C_ID;
  int32_t  s_O_ID;
  int16_t  s_O_CARRIER_ID;
  int16_t  s_ol_cnt;
  int16_t  pad1[2];
  struct   oitems_struct {
    double  s_OL_AMOUNT;
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad2;
    char     s_OL_DELIVERY_D_time[27];
  } item[15];
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_O_ENTRY_D_time[27];
  int16_t  pad3[2];
};

struct in_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_W_ID;
  int16_t  s_O_CARRIER_ID;
};

struct out_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_O_ID[10];
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

struct in_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_threshold;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
};

struct out_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_low_stock;
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

/* ******************************************************************** */
/* Transaction Prototypes                                               */
/* ******************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/* ******************************************************************** */
/* DB2 Connect/Disconnect & Thread Context Wrappers                     */
/* ******************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);


#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H
```

include/lval.h

```
/* lval.h - generated automatically at 20060905.1052 */

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 41600
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H
```

include/tpccapp.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpccapp.h - Application Macros
 */

#ifndef __TPCCAPP_H
#define __TPCCAPP_H

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>

#define daricall

#include "sqlca.h"
#include "sqlcodes.h"


#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var)  SwapEndian((void*)&Var, sizeof(Var))

/*****************************************************************************
  FUNCTION: SwapEndian
  PURPOSE: Swap the byte order of a structure
  EXAMPLE: int I=0x12345678; SWAP_BYTE(I); I => 0x78563412;
  IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op
          e.g.: *a = 0x12 [ Addr + 0];
                *b = 0x78 [ Add + 4 - 0 - 1 = Addr+3];
                *a ^= *b;         // sets *a to 0x6A
                *b ^= *a;         // sets *b to 0x12
                *a ^= *b;         // sets *a to 0x78

           Now *a => 0x78 && *b => 0x12
*****************************************************************************/

void SwapEndian(void *Addr, int nb)
{
  int i;
  for (i=0; i<nb/2; i++)
  {
    char *a = (char*)Addr+i;
    char *b = (char*)Addr+(nb-i-1);

    *a ^= *b;
    *b ^= *a;
    *a ^= *b;
  }
}
#endif   //SWAP_ENDIAN


/*****************************************************************************/
/* SQLCODE Macros                                                            */
/*****************************************************************************/

#define DLCHK(a) \
  if (sqlca.sqlcode == SQL_RC_E911) { goto a; }

#define NACOMPCHK(last) \
  if (sqlca.sqlcode != SQL_RC_E1339) { last = -1; } \
  else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : sqlca.sqlerrmc[4]-0x30); \
         int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]-0x30); \
         if (b == 0) { last = a; } else { last = a * 10 + b; } \
       }

#endif // __TPCCAPP_H
```

include/tpccdbg.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpccdbg.h - Debugging Macros
 *
 */
```

```
#ifndef __TPCCDBG_H
#define __TPCCDBG_H

#ifdef __cplusplus
extern "C" {
#endif

extern void sqlerror (int tranType, char *msg, char *file, int line,
                      SQL_STRUCTURE sqlca *psqlca);

extern void new_debug (struct out_neword_struct *neword_ptr,
                       struct in_neword_struct *in_neword_ptr,
                       char *msg);
extern void pay_debug (struct out_payment_struct *payment_ptr,
                       struct in_payment_struct *in_payment_ptr,
                       char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                       struct in_ordstat_struct *in_ordstat_ptr,
                       char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
                       struct in_delivery_struct *in_delivery_ptr,
                       char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
                       struct in_stocklev_struct *in_stocklev_ptr,
                       char *msg);

extern void new_print (struct out_neword_struct *neword_ptr,
                       struct in_neword_struct *in_neword_ptr,
                       char *filename,
                       char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
                       struct in_payment_struct *in_payment_ptr,
                       char *filename,
                       char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
                       struct in_ordstat_struct *in_ordstat_ptr,
                       char *filename,
                       char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
                       struct in_delivery_struct *in_delivery_ptr,
                       char *filename,
                       char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
                       struct in_stocklev_struct *in_stocklev_ptr,
                       char *filename,
                       char *msg);

#ifdef __cplusplus
}
#endif

#endif // __TPCCDBG_H
```

tpccenv.sh

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
######
#
# tpccenv.sh - UNIX Environment Setup
#

# The Kit Version
export TPCC_VERSION=CK060815

# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}

# The OS being used (i.e. "UNIX", "LINUX", "WINDOWS")
export PLATFORM=LINUX

# The type of make command and slash used by the OS.
# (i.e. UNIX - "/", WINDOWS - "\").
# These are referenced all over the kit.
export SLASH="/";
export MAKE=make

# Specifies whether or not to use dari stored proc's for the TPC-C driver.  Set to either
DARIVERSION or NONDARI;
#export TPCC_SPTYPE=NOSP
#export TPCC_SPTYPE=SPGENERAL2
export TPCC_SPTYPE=SPGENERAL
#export TPCC_SPTYPE=DARI2SQLDA

export DB2VERSION=v8

# The schema name is typically the SQL autorization ID (or username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}

# DB2 EE/EEE Configuration
export DB2EDITION=EE
#export DB2EDITION=EEE
export DB2NODE=0
export DB2NODES=1;      # set to the number of nodes you have.  Set to 1 for EE.

# TPCC General Configuration
export TPCC_DBNAME=TPCC
export TPCC_ROOT=${HOME}/tpc-c.ibm
export TPCC_SQLLIB=${HOME}/sqllib
export TPCC_RUNDATA=${HOME}/tpccdata

# TPCC Debug Configuration
# This is the path where all error and debug logs are placed.
# To get debugging from within the stored procedures, you must
# set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
export TPCC_DEBUGDIR=/tmp

# Specifies where stored procedures should be placed and if they should
# be fenced.
export TPCC_SPDIR=${TPCC_SQLLIB}/function
export TPCC_FENCED=NO
```

10 Appendix B: Tunable Parameters

10.1. Database Parameters

db2set.cfg.out

```
DB2LINUXAIO=YES
DB2_RESOURCE_POLICY=/home/tpcc/tpc-c.ibm/aff2_sixteen_listener.cfg
DB2_SELUDI_COMM_BUFFER=Y
DB2_USE_ALTERNATE_PAGE_CLEANING=YES
DB2_MAX_NON_TABLE_LOCKS=500
DB2_LGPAGE_BP=ON
DB2_TRUSTED_BINDIN=ON
DB2_KEEPTABLELOCK=ON
DB2_NO_FORK_CHECK=ON
DB2_ALLOCATION_SIZE=8388608
DB2_APM_PERFORMANCE=ALL
DB2_ENABLE_BUFPD=OFF
DB2_PINNED_BP=ON
DB2_SELECTIVITY=ON
DB2ASSUMEUPDATE=ON
DB2CHECKCLIENTINTERVAL=0
DB2_HASH_JOIN=OFF
DB2CHKSQLDA=OFF
DB2_COLLECT_TS_REC_INFO=false
DB2COMM=tcpip
DB2CHKPTR=OFF
```

db.cfg.out

```
     Database Configuration for Database tpcc

 Database configuration release level            = 0x0b00
 Database release level                          = 0x0b00

 Database territory                              = US
 Database code page                              = 819
 Database code set                               = ISO8859-1
 Database country/region code                    = 1
 Database collating sequence                     = IDENTITY
 Alternate collating sequence              (ALT_COLLATE) =
 Database page size                              = 4096

 Dynamic SQL Query management           (DYN_QUERY_MGMT) = DISABLE

 Discovery support for this database       (DISCOVER_DB) = ENABLE

 Restrict access                                 = NO
 Default query optimization class        (DFT_QUERYOPT) = 5
 Degree of parallelism                   (DFT_DEGREE) = 1
 Continue upon arithmetic exceptions   (DFT_SQLMATHWARN) = NO
 Default refresh age                     (DFT_REFRESH_AGE) = 0
 Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM
 Number of frequent values retained     (NUM_FREQVALUES) = 10
 Number of quantiles retained           (NUM_QUANTILES) = 20

 Backup pending                                  = NO

 Database is consistent                          = YES
 Rollforward pending                             = NO
 Restore pending                                 = NO

 Multi-page file allocation enabled              = YES

 Log retain for recovery status                  = RECOVERY
 User exit for logging status                    = NO

 Self tuning memory                (SELF_TUNING_MEM) = OFF
 Size of database shared memory (4KB)  (DATABASE_MEMORY) = 30760536
 Database memory threshold             (DB_MEM_THRESH) = 10
 Max storage for lock list (4KB)           (LOCKLIST) = 8000
 Percent. of lock lists per application    (MAXLOCKS) = 100
 Package cache size (4KB)                (PCKCACHESZ) = 1000
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 5000
 Sort list heap (4KB)                     (SORTHEAP) = 16

 Database heap (4KB)                       (DBHEAP) = 8192
 Catalog cache size (4KB)          (CATALOGCACHE_SZ) = (MAXAPPLS*4)
 Log buffer size (4KB)                   (LOGBUFSZ) = 3000
 Utilities heap size (4KB)           (UTIL_HEAP_SZ) = 5000
 Buffer pool size (pages)                 (BUFFPAGE) = 1000
 Max size of appl. group mem set (4KB) (APPGROUP_MEM_SZ) = 30000
 Percent of mem for appl. group heap   (GROUPHEAP_RATIO) = 70
 Max appl. control heap size (4KB)     (APP_CTL_HEAP_SZ) = 128

 SQL statement heap (4KB)                  (STMTHEAP) = 16384
 Default application heap (4KB)          (APPLHEAPSZ) = 328
 Statistics heap size (4KB)           (STAT_HEAP_SZ) = 4384

 Interval for checking deadlock (ms)        (DLCHKTIME) = 3000
 Lock timeout (sec)                      (LOCKTIMEOUT) = -1

 Changed pages threshold             (CHNGPGS_THRESH) = 99
 Number of asynchronous page cleaners   (NUM_IOCLEANERS) = 1
 Number of I/O servers                  (NUM_IOSERVERS) = 1
 Index sort flag                        (INDEXSORT) = YES
 Sequential detect flag                 (SEQDETECT) = NO
 Default prefetch size (pages)        (DFT_PREFETCH_SZ) = AUTOMATIC

 Track modified pages                      (TRACKMOD) = OFF

 Default number of containers                    = 1
 Default tablespace extentsize (pages)   (DFT_EXTENT_SZ) = 32

 Max number of active applications          (MAXAPPLS) = 1400
 Average number of active applications       (AVG_APPLS) = 1
 Max DB files open per application          (MAXFILOP) = 1600

 Log file size (4KB)                     (LOGFILSIZ) = 512000
 Number of primary log files              (LOGPRIMARY) = 256
 Number of secondary log files            (LOGSECOND) = 0
 Changed path to log files                (NEWLOGPATH) =
 Path to log files                               = /dev/raw/raw1300
 Overflow log path                  (OVERFLOWLOGPATH) =
 Mirror log path                    (MIRRORLOGPATH) =
 First active log file                           = S0000116.LOG
 Block log on disk full             (BLK_LOG_DSK_FUL) = NO
 Percent max primary log space by transaction  (MAX_LOG) = 0
 Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

 Group commit count                        (MINCOMMIT) = 1
 Percent log file reclaimed before soft chckpt (SOFTMAX) = 1726
 Log retain for recovery enabled           (LOGRETAIN) = RECOVERY
 User exit for logging enabled             (USEREXIT) = OFF

 HADR database role                              = STANDARD
 HADR local host name                (HADR_LOCAL_HOST) =
 HADR local service name              (HADR_LOCAL_SVC) =
 HADR remote host name               (HADR_REMOTE_HOST) =
 HADR remote service name             (HADR_REMOTE_SVC) =
 HADR instance name of remote server  (HADR_REMOTE_INST) =
 HADR timeout value                        (HADR_TIMEOUT) = 120
 HADR log write synchronization mode     (HADR_SYNCMODE) = NEARSYNC

 First log archive method               (LOGARCHMETH1) = LOGRETAIN
 Options for logarchmeth1                 (LOGARCHOPT1) =
 Second log archive method                (LOGARCHMETH2) = OFF
 Options for logarchmeth2                 (LOGARCHOPT2) =
 Failover log archive path                (FAILARCHPATH) =
 Number of log archive retries on error   (NUMARCHRETRY) = 5
 Log archive retry Delay (secs)           (ARCHRETRYDELAY) = 20
 Vendor options                           (VENDOROPT) =

 Auto restart enabled                      (AUTORESTART) = ON
 Index re-creation time and redo index build  (INDEXREC) = SYSTEM (RESTART)
 Log pages during index build             (LOGINDEXBUILD) = OFF
 Default number of loadrec sessions    (DFT_LOADREC_SES) = 1
 Number of database backups to retain   (NUM_DB_BACKUPS) = 12
 Recovery history retention (days)     (REC_HIS_RETENTN) = 366

 TSM management class                    (TSM_MGMTCLASS) =
 TSM node name                           (TSM_NODENAME) =
 TSM owner                               (TSM_OWNER) =
 TSM password                            (TSM_PASSWORD) =

 Automatic maintenance                     (AUTO_MAINT) = OFF
   Automatic database backup               (AUTO_DB_BACKUP) = OFF
   Automatic table maintenance             (AUTO_TBL_MAINT) = OFF
     Automatic runstats                    (AUTO_RUNSTATS) = OFF
     Automatic statistics profiling        (AUTO_STATS_PROF) = OFF
       Automatic profile updates           (AUTO_PROF_UPD) = OFF
     Automatic reorganization              (AUTO_REORG) = OFF
```

dbm.cfg.out

```
        Database Manager Configuration

    Node type = Enterprise Server Edition with local and remote clients

 Database manager configuration release level            = 0x0b00

 CPU speed (millisec/instruction)             (CPUSPEED) = 1.889377e-07
 Communications bandwidth (MB/sec)      (COMM_BANDWIDTH) = 1.000000e+02

 Max number of concurrently active databases     (NUMDB) = 1
 Federated Database System Support          (FEDERATED) = NO
 Transaction processor monitor name       (TP_MON_NAME) =

 Default charge-back account          (DFT_ACCOUNT_STR) =

 Java Development Kit installation path       (JDK_PATH) = /home/tpcc/sqllib/java/jdk64

 Diagnostic error capture level             (DIAGLEVEL) = 1
 Notify Level                            (NOTIFYLEVEL) = 1
 Diagnostic data directory path             (DIAGPATH) =

 Default database monitor switches
   Buffer pool                  (DFT_MON_BUFPOOL) = OFF
   Lock                         (DFT_MON_LOCK) = OFF
   Sort                         (DFT_MON_SORT) = OFF
   Statement                    (DFT_MON_STMT) = OFF
   Table                        (DFT_MON_TABLE) = OFF
   Timestamp                    (DFT_MON_TIMESTAMP) = OFF
   Unit of work                 (DFT_MON_UOW) = OFF
 Monitor health of instance and databases   (HEALTH_MON) = OFF

 SYSADM group name                       (SYSADM_GROUP) =
 SYSCTRL group name                      (SYSCTRL_GROUP) =
 SYSMAINT group name                     (SYSMAINT_GROUP) =
 SYSMON group name                       (SYSMON_GROUP) =
```

```
Client Userid-Password Plugin          (CLNT_PW_PLUGIN) =
Client Kerberos Plugin                 (CLNT_KRB_PLUGIN) =
Group Plugin                           (GROUP_PLUGIN) =
GSS Plugin for Local Authorization     (LOCAL_GSSPLUGIN) =
Server Plugin Mode                     (SRV_PLUGIN_MODE) = UNFENCED
Server List of GSS Plugins             (SRVCON_GSSPLUGIN_LIST) =
Server Userid-Password Plugin          (SRVCON_PW_PLUGIN) =
Server Connection Authentication       (SRVCON_AUTH) = NOT_SPECIFIED
Database manager authentication        (AUTHENTICATION) = CLIENT
Cataloging allowed without authority   (CATALOG_NOAUTH) = NO
Trust all clients                      (TRUST_ALLCLNTS) = YES
Trusted client authentication          (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication        (FED_NOAUTH) = NO

Default database path                  (DFTDBPATH) = /home/tpcc

Database monitor heap size (4KB)       (MON_HEAP_SZ) = 4096
Java Virtual Machine heap size (4KB)   (JAVA_HEAP_SZ) = 1024
Audit buffer size (4KB)                (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB)   (INSTANCE_MEMORY) = AUTOMATIC
Backup buffer default size (4KB)       (BACKBUFSZ) = 1024
Restore buffer default size (4KB)      (RESTBUFSZ) = 1024

Sort heap threshold (4KB)              (SHEAPTHRES) = 0

Directory cache support                (DIR_CACHE) = YES

Application support layer heap size (4KB)   (ASLHEAPSZ) = 15
Max requester I/O block size (bytes)        (RQRIOBLK) = 4096
Query heap size (4KB)                  (QUERY_HEAP_SZ) = 1000

Workload impact by throttled utilities(UTIL_IMPACT_LIM) = 10

Priority of agents                     (AGENTPRI) = SYSTEM
Max number of existing agents          (MAXAGENTS) = 1400
Agent pool size                        (NUM_POOLAGENTS) = 0
Initial number of agents in pool       (NUM_INITAGENTS) = 0
Max number of coordinating agents      (MAX_COORDAGENTS) = (MAXAGENTS -
NUM_INITAGENTS)
Max no. of concurrent coordinating agents  (MAXCAGENTS) = MAX_COORDAGENTS
Max number of client connections       (MAX_CONNECTIONS) = MAX_COORDAGENTS

Keep fenced process                    (KEEPFENCED) = YES
Number of pooled fenced processes      (FENCED_POOL) = MAX_COORDAGENTS
Initial number of fenced processes     (NUM_INITFENCED) = 0

Index re-creation time and redo index build  (INDEXREC) = RESTART

Transaction manager database name      (TM_DATABASE) = 1ST_CONN
Transaction resync interval (sec)      (RESYNC_INTERVAL) = 180

SPM name                               (SPM_NAME) =
SPM log size                           (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit                 (SPM_MAX_RESYNC) = 20
SPM log path                           (SPM_LOG_PATH) =

TCP/IP Service name                    (SVCENAME) =
Discovery mode                         (DISCOVER) = SEARCH
Discover server instance               (DISCOVER_INST) = ENABLE

Maximum query degree of parallelism    (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism     (INTRA_PARALLEL) = NO

Maximum Asynchronous TQs per query     (FEDERATED_ASYNC) = 0

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) = AUTOMATIC
No. of int. communication channels     (FCM_NUM_CHANNELS) = AUTOMATIC
Node connection elapse time (sec)      (CONN_ELAPSE) = 10
Max number of node connection retries (MAX_CONNRETRIES) = 5
Max time difference between nodes (min) (MAX_TIME_DIFF) = 60

db2start/db2stop timeout (min)         (START_STOP_TIME) = 10
```

aff2_sixteen_listener.cfg

```
<!-- This policy is valid -->
<RESOURCE_POLICY>
  <DATABASE_RESOURCE_POLICY>
    <DBNAME>tpcc</DBNAME>
    <METHOD>NODEMASK</METHOD>

    <RESOURCE_BINDING>
      <RESOURCE>0</RESOURCE>
      <DBMEM_PERCENTAGE>49.5</DBMEM_PERCENTAGE>

      <SERVICE_NAME>50021</SERVICE_NAME>
      <SERVICE_NAME>50022</SERVICE_NAME>
      <SERVICE_NAME>50023</SERVICE_NAME>
      <SERVICE_NAME>50024</SERVICE_NAME>
      <SERVICE_NAME>50025</SERVICE_NAME>
      <SERVICE_NAME>50026</SERVICE_NAME>
      <SERVICE_NAME>50027</SERVICE_NAME>
      <SERVICE_NAME>50028</SERVICE_NAME>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>5</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>21</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>37</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>53</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>69</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>85</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>101</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>6</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>22</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>38</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>54</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>70</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>86</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>102</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>7</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>23</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>39</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>55</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>71</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>87</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>103</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>8</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>24</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>40</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>56</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>72</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>88</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>104</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>9</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>25</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>41</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>57</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>73</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>89</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>105</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>10</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>26</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>42</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>58</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>74</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>90</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>106</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>11</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>27</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>43</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>59</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>75</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>91</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>107</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>12</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>28</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>44</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>60</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>76</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>92</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>108</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
    </RESOURCE_BINDING>

    <RESOURCE_BINDING>
      <RESOURCE>1</RESOURCE>
      <DBMEM_PERCENTAGE>49.5</DBMEM_PERCENTAGE>

      <SERVICE_NAME>50029</SERVICE_NAME>
      <SERVICE_NAME>50030</SERVICE_NAME>
      <SERVICE_NAME>50031</SERVICE_NAME>
      <SERVICE_NAME>50032</SERVICE_NAME>
      <SERVICE_NAME>50033</SERVICE_NAME>
      <SERVICE_NAME>50034</SERVICE_NAME>
      <SERVICE_NAME>50035</SERVICE_NAME>
      <SERVICE_NAME>50036</SERVICE_NAME>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>13</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>29</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>45</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>61</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>77</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>93</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>109</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>14</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>30</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>46</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>62</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>78</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>94</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>110</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>1</NUM_CLEANERS>
        <BUFFERPOOL_ID>15</BUFFERPOOL_ID>
```

```
      <BUFFERPOOL_ID>31</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>47</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>63</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>79</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>95</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>111</BUFFERPOOL_ID>
    </BUFFERPOOL_BINDING>
    <BUFFERPOOL_BINDING>
      <NUM_CLEANERS>1</NUM_CLEANERS>
      <BUFFERPOOL_ID>16</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>32</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>48</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>64</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>80</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>96</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>112</BUFFERPOOL_ID>
    </BUFFERPOOL_BINDING>
    <BUFFERPOOL_BINDING>
      <NUM_CLEANERS>1</NUM_CLEANERS>
      <BUFFERPOOL_ID>17</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>33</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>49</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>65</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>81</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>97</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>113</BUFFERPOOL_ID>
    </BUFFERPOOL_BINDING>
    <BUFFERPOOL_BINDING>
      <NUM_CLEANERS>1</NUM_CLEANERS>
      <BUFFERPOOL_ID>18</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>34</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>50</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>66</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>82</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>98</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>114</BUFFERPOOL_ID>
    </BUFFERPOOL_BINDING>
    <BUFFERPOOL_BINDING>
      <NUM_CLEANERS>1</NUM_CLEANERS>
      <BUFFERPOOL_ID>19</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>35</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>51</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>67</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>83</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>99</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>115</BUFFERPOOL_ID>
    </BUFFERPOOL_BINDING>
    <BUFFERPOOL_BINDING>
      <NUM_CLEANERS>1</NUM_CLEANERS>
      <BUFFERPOOL_ID>20</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>36</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>52</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>68</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>84</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>100</BUFFERPOOL_ID>
      <BUFFERPOOL_ID>116</BUFFERPOOL_ID>
    </BUFFERPOOL_BINDING>
  </RESOURCE_BINDING>

 </DATABASE_RESOURCE_POLICY>
</RESOURCE_POLICY>
```

10.2. Transaction Monitor Parameters

inetInfo_registry.reg

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000040
"DispatchEntries"=hex(7):00,00,13,04,9d,3d,83,7c,40,9d,88,7c,cc,8b,13,04,00,00,\
  00,00
"PoolThreadLimit"=dword:00000320
"MaxPoolThreads"=dword:000000c8
"MaxConcurrency"=dword:00000320
```

tcpip_parameters_registery.reg

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="client2"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\
  00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\
  64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="client2"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"EnableTCPA"=dword:00000001
"EnableRSS"=dword:00000001
"EnableTCPChimney"=dword:00000001
"SynAttackProtect"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
  6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
  00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00,30,00,45,00,35,00,37,00,35,00,\
  38,00,43,00,2d,00,46,00,35,00,34,00,46,00,2d,00,34,00,43,00,31,00,45,00,2d,\
  00,41,00,45,00,46,00,36,00,2d,00,38,00,37,00,32,00,37,00,30,00,34,00,41,00,\
  43,00,41,00,36,00,38,00,42,00,7d,00,00,00,54,00,63,00,70,00,69,00,70,00,5c,\
  00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,\
  6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,00,38,00,30,\
  00,32,00,43,00,41,00,37,00,38,00,45,00,2d,00,34,00,38,00,32,00,35,00,2d,00,\
  34,00,34,00,42,00,38,00,2d,00,39,00,36,00,32,00,36,00,2d,00,30,00,31,00,34,\
  00,32,00,33,00,34,00,32,00,39,00,32,00,32,00,46,00,34,00,7d,00,00,00,00,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:8c,75,e5,20,4f,f5,1e,4c,ae,f6,87,27,04,ac,a6,8b,8e,a7,2c,80,\
  25,48,b8,44,96,26,01,42,34,29,22,f4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{2C482B80-E81F-4472-83B2-C606043D4069}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
  6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
  00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00,43,00,34,00,38,00,32,00,42,00,\
  38,00,30,00,2d,00,45,00,38,00,31,00,46,00,2d,00,34,00,34,00,37,00,32,00,2d,\
  00,38,00,33,00,42,00,32,00,2d,00,43,00,36,00,30,00,36,00,30,00,34,00,33,00,\
  44,00,34,00,30,00,36,00,39,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{72844CD3-4B9F-4A3E-8169-F6771A71D2D0}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
  6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
  00,61,00,63,00,65,00,73,00,5c,00,7b,00,37,00,32,00,38,00,34,00,34,00,43,00,\
  44,00,33,00,2d,00,34,00,42,00,39,00,46,00,2d,00,34,00,41,00,33,00,45,00,2d,\
  00,38,00,31,00,36,00,39,00,2d,00,46,00,36,00,37,00,37,00,31,00,41,00,37,00,\
  31,00,44,00,32,00,44,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DNSRegisteredAdapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{20E5758C-F54F-4C1E-AEF6-872704ACA68B}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{2C482B80-E81F-4472-83B2-C606043D4069}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,36,00,38,00,2e,00,31,00,31,00,\
  2e,00,31,00,30,00,32,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
  00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
  32,00,00,00,00,00
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:45883c95
"T1"=dword:4588439d
"T2"=dword:458848e3
"LeaseTerminatesTime"=dword:45884aa5
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{72844CD3-4B9F-4A3E-8169-F6771A71D2D0}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,32,00,2e,00,31,00,2e,00,32,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
  00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
```

33,00,00,00,00,00
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:45883c79
"T1"=dword:45884381
"T2"=dword:458848c7
"LeaseTerminatesTime"=dword:45884a89
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"TcpWindowSize"=dword:00040000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{802CA78E-4825-44B8-9626-0142342922F4}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,\
  6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,\
  00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,00,6c,00,6c,00,\
  00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,00,02,\
  00,00,00,01,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00,00,\
  00,00,00,00,00,00,06,00,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00,00,\
  00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,02,00,00,00,\
  00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00,02,\
  00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00,00

tpccCom.tpcc_com_settingss.txt.txt

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"dbInterfacePath"="C:\\inetpub\\wwwroot\\tpcc\\tpccdb2glue.dll"
"dbName"="tpcc"
"dbPassword"="tpcc"
"dbType"="DB2"
"dbUserName"="tpcc"
"dlvyLogPath"="C:\\inetpub\\wwwroot\\tpcc"
"dlvyQueueLen"=dword:00004e20
"dlvyThreads"=dword:00000007
"errorLogFile"="c:\\inetpub\\wwwroot\\tpcc\\isapi_err.log"
"htmlTrace"=dword:00000000
"htmlTraceLogFile"="c:\\inetpub\\wwwroot\\tpcc\\isapi.log"
"isapi_trace"=dword:00000000
"nullDB"=dword:00000000
"numPools"=dword:00000001
"numServers"=dword:00000001
"numUsers"=dword:00007530
"numWarehouse"=dword:000007d0

tpcc_software_registry.reg client1

Transactions not supported
Enable Object pooling
Minimum pool size 72
Maximum pool size 72
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client2

Transactions not supported
Enable Object pooling
Minimum pool size 69
Maximum pool size 69
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client3

Transactions not supported
Enable Object pooling
Minimum pool size 71
Maximum pool size 71
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client4

Transactions not supported
Enable Object pooling
Minimum pool size 72
Maximum pool size 72
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client5

Transactions not supported
Enable Object pooling
Minimum pool size 72
Maximum pool size 72
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client6

Transactions not supported
Enable Object pooling
Minimum pool size 71
Maximum pool size 71
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client7

Transactions not supported
Enable Object pooling
Minimum pool size 70
Maximum pool size 70
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client8

Transactions not supported
Enable Object pooling
Minimum pool size 71
Maximum pool size 71
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client9

Transactions not supported
Enable Object pooling
Minimum pool size 70
Maximum pool size 70
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client10

Transactions not supported
Enable Object pooling
Minimum pool size 69
Maximum pool size 69
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client11

Transactions not supported
Enable Object pooling
Minimum pool size 70
Maximum pool size 70
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client12

Transactions not supported
Enable Object pooling
Minimum pool size 69
Maximum pool size 69
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client13

Transactions not supported
Enable Object pooling
Minimum pool size 69
Maximum pool size 69
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client14

Transactions not supported
Enable Object pooling
Minimum pool size 71
Maximum pool size 71
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc_software_registry.reg client15

```
Transactions not supported
Enable Object pooling
Minimum pool size 71
Maximum pool size 71
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required
```

tpcc_software_registry.reg client16

```
Transactions not supported
Enable Object pooling
Minimum pool size 70
Maximum pool size 70
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required
```

10.3. Linux Parameters

alloc_hugepages.sh

```
#!/bin/bash

echo 60100 > /proc/sys/vm/nr_hugepages
```

cmdline.txt

```
root=/dev/sda2 resume=/dev/sda1 selinux=0 audit=0 console=ttyS0,115200 elevator=noop
splash=silent  showopts migration_cost=-1000000,1610,13327,24669
```

chrt-ififo2.sh

```
#!/bin/ksh

    for i in `ps -fu tpcc | grep db2logg| awk '{print $2}'`; do
        chrt -f -p 97 $i
    done

    for j in `ps -fu tpcc | grep db2ag | awk '{print $2}'`; do
        chrt -f -p 96 $j
    done

    for k in `ps -fu tpcc | grep db2pclnr | awk '{print $2}'`; do
        chrt -f -p 96 $k
    done
```

zio.sh

```
echo 1000 > /sys/class/scsi_host/host24/zio_timer
echo 1000 > /sys/class/scsi_host/host25/zio_timer
echo 1000 > /sys/class/scsi_host/host26/zio_timer
echo 1000 > /sys/class/scsi_host/host27/zio_timer
echo 1000 > /sys/class/scsi_host/host28/zio_timer
echo 1000 > /sys/class/scsi_host/host29/zio_timer
echo 1000 > /sys/class/scsi_host/host30/zio_timer
echo 1000 > /sys/class/scsi_host/host31/zio_timer
echo 1000 > /sys/class/scsi_host/host32/zio_timer
echo 1000 > /sys/class/scsi_host/host33/zio_timer
echo 1000 > /sys/class/scsi_host/host34/zio_timer
echo 1000 > /sys/class/scsi_host/host35/zio_timer
echo 1000 > /sys/class/scsi_host/host36/zio_timer
echo 1000 > /sys/class/scsi_host/host37/zio_timer
echo 1000 > /sys/class/scsi_host/host38/zio_timer
echo 1000 > /sys/class/scsi_host/host39/zio_timer
echo 1000 > /sys/class/scsi_host/host40/zio_timer
echo 1000 > /sys/class/scsi_host/host41/zio_timer
echo 1000 > /sys/class/scsi_host/host42/zio_timer
echo 1000 > /sys/class/scsi_host/host43/zio_timer

echo 1 > /sys/class/scsi_host/host24/zio
echo 1 > /sys/class/scsi_host/host25/zio
echo 1 > /sys/class/scsi_host/host26/zio
echo 1 > /sys/class/scsi_host/host27/zio
echo 1 > /sys/class/scsi_host/host28/zio
echo 1 > /sys/class/scsi_host/host29/zio
echo 1 > /sys/class/scsi_host/host30/zio
echo 1 > /sys/class/scsi_host/host31/zio
echo 1 > /sys/class/scsi_host/host32/zio
echo 1 > /sys/class/scsi_host/host33/zio
echo 1 > /sys/class/scsi_host/host34/zio
echo 1 > /sys/class/scsi_host/host35/zio
echo 1 > /sys/class/scsi_host/host36/zio
echo 1 > /sys/class/scsi_host/host37/zio
echo 1 > /sys/class/scsi_host/host38/zio
echo 1 > /sys/class/scsi_host/host39/zio
echo 1 > /sys/class/scsi_host/host40/zio
echo 1 > /sys/class/scsi_host/host41/zio
echo 1 > /sys/class/scsi_host/host42/zio
echo 1 > /sys/class/scsi_host/host43/zio
```

aff1.sh

```
#!/bin/ksh

to_hex() {
        if (($# == 0)); then
                print "usage: to_hex <int> [<size>]"
                return
        fi

        let int="${1}"
        let size="${2:-0}"      # length of output hex string.

        set -A digit 0 1 2 3 4 5 6 7 8 9 a b c d e f
        hex=''

        while ((int > 0)); do
                hex="${digit[$((int % 16))]}${hex}"
                int=$((int / 16))
        done

        if ((size != 0)); then
                typeset -Z${size} zeros=0
                typeset -R${size} hex="${zeros}${hex}"
        fi

#       print "${hex}"
}

for i in `ps -fu tpcc | grep db2ipccm | awk '{print $2}'`; do
                cmd="taskset -p 0x0001 $i"
                $cmd
done

typeset -i j
j=3
k=3

for i in `ps -fu tpcc | grep db2tcpcm | awk '{print $2}'`; do
                if (($j > 805306368)); then
                        hex=c0000000
                else
                        to_hex $j
                fi
                cmd="taskset -p 0x${hex} $i"
                $cmd
                j=($j*4);
done
```

aff2.sh

```
#!/bin/ksh

to_hex() {
        if (($# == 0)); then
                print "usage: to_hex <int> [<size>]"
                return
        fi

        let int="${1}"
        let size="${2:-0}"      # length of output hex string.

        set -A digit 0 1 2 3 4 5 6 7 8 9 a b c d e f
        hex=''

        while ((int > 0)); do
                hex="${digit[$((int % 16))]}${hex}"
                int=$((int / 16))
        done

        if ((size != 0)); then
                typeset -Z${size} zeros=0
                typeset -R${size} hex="${zeros}${hex}"
        fi

#       print "${hex}"
}

for i in `ps -fu tpcc | grep db2loggw | awk '{print $2}'`; do
                cmd="taskset -p 0xffff0000 $i"
                $cmd
done

j=3221225472
print "Cleaners:"

for i in `ps -fu tpcc | grep db2pclnr | awk '{print $2}'`; do
                if (($j > 805306368)); then
                        hex=c0000000
                else
                        to_hex $j
                fi
                if (($j > 0)); then
                        cmd="taskset -p 0x${hex} $i"
                        $cmd
                        j=$j/4;
                else
                        echo "not changing affinity for pid $i"
                fi
done
```

affinitize_irqs.sh

```
#!/bin/bash

# chassis 1 - CPU's 0-15
```

```
# database storage - qlogic 4Gb ports
echo 00000000,00000000,00000000,00000002 > /proc/irq/177/smp_affinity  # CPU 1
echo 00000000,00000000,00000000,00000004 > /proc/irq/185/smp_affinity  # CPU 2
echo 00000000,00000000,00000000,00000010 > /proc/irq/193/smp_affinity  # CPU 4
echo 00000000,00000000,00000000,00000020 > /proc/irq/201/smp_affinity  # CPU 5
echo 00000000,00000000,00000000,00000080 > /proc/irq/209/smp_affinity  # CPU 7
echo 00000000,00000000,00000000,00000100 > /proc/irq/217/smp_affinity  # CPU 8
# database storage - qlogic 2Gb ports
echo 00000000,00000000,00000000,00000800 > /proc/irq/225/smp_affinity  # CPU 11
echo 00000000,00000000,00000000,00001000 > /proc/irq/233/smp_affinity  # CPU 12
echo 00000000,00000000,00000000,00004000 > /proc/irq/50/smp_affinity   # CPU 14
echo 00000000,00000000,00000000,00008000 > /proc/irq/58/smp_affinity   # CPU 15
# flat files - Emulex ports - just choose some unused CPU's
echo 00000000,00000000,00000000,00000200 > /proc/irq/154/smp_affinity  # CPU 9
echo 00000000,00000000,00000000,00000400 > /proc/irq/162/smp_affinity  # CPU 10

# chassis 2 - CPU's 16-31

# database storage - qlogic 4Gb ports
echo 00000000,00000000,00000000,00040000 > /proc/irq/66/smp_affinity   # CPU 18
echo 00000000,00000000,00000000,00080000 > /proc/irq/74/smp_affinity   # CPU 19
echo 00000000,00000000,00000000,00100000 > /proc/irq/82/smp_affinity   # CPU 20
echo 00000000,00000000,00000000,00200000 > /proc/irq/90/smp_affinity   # CPU 21
echo 00000000,00000000,00000000,00800000 > /proc/irq/98/smp_affinity   # CPU 23
echo 00000000,00000000,00000000,01000000 > /proc/irq/106/smp_affinity  # CPU 24
# database storage - qlogic 2Gb ports
echo 00000000,00000000,00000000,04000000 > /proc/irq/114/smp_affinity  # CPU 26
echo 00000000,00000000,00000000,08000000 > /proc/irq/122/smp_affinity  # CPU 27
echo 00000000,00000000,00000000,10000000 > /proc/irq/130/smp_affinity  # CPU 28
echo 00000000,00000000,00000000,20000000 > /proc/irq/138/smp_affinity  # CPU 29
# log - qlogic 2Gb port
echo 00000000,00000000,00000000,80000000 > /proc/irq/146/smp_affinity  # CPU 31

/home/tpcc/affirq.sh
```

affirq.sh

```
#!/bin/ksh

# 192.168.11 network - top chassis
device=`cat /proc/interrupts | grep eth9 | awk -F : '{print $1}'`
print "eth9: echo 00000000,0000000,00000000,0002000 > /proc/irq/$device/smp_affinity"
echo "00000000,0000000,00000000,00002000" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity

# 192.168.12 network - bottom chassis
device=`cat /proc/interrupts | grep eth11 | awk -F : '{print $1}'`
print "eth11: echo 00000000,0000000,00000000,00010000 >
/proc/irq/$device/smp_affinity"
echo "00000000,0000000,00000000,00010000" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity
```

doit

```
# !/bin/sh

# tune slab cache
/home/tpcc/tune_slab.sh

echo "affinitizing irqs"
/home/tpcc/tpc-c.ibm/affinitize_irqs.sh

sysctl -p
modprobe raw
sleep 5
/home/tpcc/tpc-c.ibm/setraw.sh

chmod -R 777 /dev/raw/raw*

modprobe capability disable=1
modprobe capability enable=1

umount  /sys/kernel/debug
umount /sys/kernel/security

/home/tpcc/tpc-c.ibm/seteth.sh
```

tune_slab.sh

```
#!/bin/bash

echo "blkdev_requests 336 168 8" > /proc/slabinfo
echo "size-1024       216 108 8" > /proc/slabinfo
echo "scsi_cmd_cache  336 168 8" > /proc/slabinfo
echo "size-4096       168  84 8" > /proc/slabinfo
echo "sgpool-8        240 120 8" > /proc/slabinfo
echo "bio             240 120 8" > /proc/slabinfo
echo "kiocb           240 120 8" > /proc/slabinfo
echo "biovec-1        240 120 8" > /proc/slabinfo
```

/etc/sysctl.conf

```
# Disable response to broadcasts.
# You don't want yourself becoming a Smurf amplifier.
net.ipv4.icmp_echo_ignore_broadcasts = 1
# enable route verification on all interfaces
net.ipv4.conf.all.rp_filter = 1
kernel.sem = 250        256000  32      1024
kernel.msgmni = 2048
kernel.shmmax = 1073741824
kernel.shmall = 137438953472
fs.file-max = 524288
net.core.rmem_max = 131071
net.core.wmem_max = 131071
vm.hugetlb_shm_group = 102
```

setraw.sh

```
# !/bin/sh
modprobe raw

raw /dev/raw/raw1    /dev/sdb1
raw /dev/raw/raw2    /dev/sdc1
raw /dev/raw/raw3    /dev/sdd1
raw /dev/raw/raw4    /dev/sde1
raw /dev/raw/raw5    /dev/sdf1
raw /dev/raw/raw6    /dev/sdg1
raw /dev/raw/raw7    /dev/sdh1
raw /dev/raw/raw8    /dev/sdi1
raw /dev/raw/raw9    /dev/sdj1
raw /dev/raw/raw10    /dev/sdk1
raw /dev/raw/raw11    /dev/sdl1
raw /dev/raw/raw12    /dev/sdm1
raw /dev/raw/raw13    /dev/sdn1
raw /dev/raw/raw14    /dev/sdo1
raw /dev/raw/raw15    /dev/sdp1
raw /dev/raw/raw16    /dev/sdq1
raw /dev/raw/raw17    /dev/sdr1
raw /dev/raw/raw18    /dev/sds1
raw /dev/raw/raw19    /dev/sdt1
raw /dev/raw/raw20    /dev/sdu1
raw /dev/raw/raw21    /dev/sdv1
raw /dev/raw/raw22    /dev/sdw1
raw /dev/raw/raw23    /dev/sdx1
raw /dev/raw/raw24    /dev/sdy1
raw /dev/raw/raw25    /dev/sdz1
raw /dev/raw/raw26    /dev/sdaa1
raw /dev/raw/raw27    /dev/sdab1
raw /dev/raw/raw28    /dev/sdac1
raw /dev/raw/raw29    /dev/sdad1
raw /dev/raw/raw30    /dev/sdae1
raw /dev/raw/raw31    /dev/sdaf1
raw /dev/raw/raw32    /dev/sdag1
raw /dev/raw/raw33    /dev/sdah1
raw /dev/raw/raw34    /dev/sdai1
raw /dev/raw/raw35    /dev/sdaj1
raw /dev/raw/raw36    /dev/sdak1
raw /dev/raw/raw37    /dev/sdal1
raw /dev/raw/raw38    /dev/sdam1
raw /dev/raw/raw39    /dev/sdan1
raw /dev/raw/raw40    /dev/sdao1
raw /dev/raw/raw41    /dev/sdap1
raw /dev/raw/raw42    /dev/sdaq1
raw /dev/raw/raw43    /dev/sdar1
raw /dev/raw/raw44    /dev/sdas1
raw /dev/raw/raw45    /dev/sdat1
raw /dev/raw/raw46    /dev/sdau1
raw /dev/raw/raw47    /dev/sdav1
raw /dev/raw/raw48    /dev/sdaw1
raw /dev/raw/raw49    /dev/sdax1
raw /dev/raw/raw50    /dev/sday1
raw /dev/raw/raw51    /dev/sdaz1
raw /dev/raw/raw52    /dev/sdba1
raw /dev/raw/raw53    /dev/sdbb1
raw /dev/raw/raw54    /dev/sdbc1
raw /dev/raw/raw55    /dev/sdbd1
raw /dev/raw/raw56    /dev/sdbe1
raw /dev/raw/raw57    /dev/sdbf1
raw /dev/raw/raw58    /dev/sdbg1
raw /dev/raw/raw59    /dev/sdbh1
raw /dev/raw/raw60    /dev/sdbi1
raw /dev/raw/raw61    /dev/sdbj1
raw /dev/raw/raw62    /dev/sdbk1
raw /dev/raw/raw63    /dev/sdbl1
raw /dev/raw/raw64    /dev/sdbm1
raw /dev/raw/raw65    /dev/sdbn1
raw /dev/raw/raw66    /dev/sdbo1
raw /dev/raw/raw67    /dev/sdbp1
raw /dev/raw/raw68    /dev/sdbq1
raw /dev/raw/raw69    /dev/sdbr1
raw /dev/raw/raw70    /dev/sdbs1
raw /dev/raw/raw71    /dev/sdbt1
raw /dev/raw/raw72    /dev/sdbu1
raw /dev/raw/raw73    /dev/sdbv1
raw /dev/raw/raw74    /dev/sdbw1
raw /dev/raw/raw75    /dev/sdbx1
raw /dev/raw/raw76    /dev/sdby1
raw /dev/raw/raw77    /dev/sdbz1
raw /dev/raw/raw78    /dev/sdca1
raw /dev/raw/raw79    /dev/sdcb1
raw /dev/raw/raw80    /dev/sdcc1

raw /dev/raw/raw81    /dev/sdb2
raw /dev/raw/raw82    /dev/sdc2
raw /dev/raw/raw83    /dev/sdd2
raw /dev/raw/raw84    /dev/sde2
raw /dev/raw/raw85    /dev/sdf2
raw /dev/raw/raw86    /dev/sdg2
raw /dev/raw/raw87    /dev/sdh2
raw /dev/raw/raw88    /dev/sdi2
raw /dev/raw/raw89    /dev/sdj2
```

raw /dev/raw/raw90 /dev/sdk2
raw /dev/raw/raw91 /dev/sdl2
raw /dev/raw/raw92 /dev/sdm2
raw /dev/raw/raw93 /dev/sdn2
raw /dev/raw/raw94 /dev/sdo2
raw /dev/raw/raw95 /dev/sdp2
raw /dev/raw/raw96 /dev/sdq2
raw /dev/raw/raw97 /dev/sdr2
raw /dev/raw/raw98 /dev/sds2
raw /dev/raw/raw99 /dev/sdt2
raw /dev/raw/raw100 /dev/sdu2
raw /dev/raw/raw101 /dev/sdv2
raw /dev/raw/raw102 /dev/sdw2
raw /dev/raw/raw103 /dev/sdx2
raw /dev/raw/raw104 /dev/sdy2
raw /dev/raw/raw105 /dev/sdz2
raw /dev/raw/raw106 /dev/sdaa2
raw /dev/raw/raw107 /dev/sdab2
raw /dev/raw/raw108 /dev/sdac2
raw /dev/raw/raw109 /dev/sdad2
raw /dev/raw/raw110 /dev/sdae2
raw /dev/raw/raw111 /dev/sdaf2
raw /dev/raw/raw112 /dev/sdag2
raw /dev/raw/raw113 /dev/sdah2
raw /dev/raw/raw114 /dev/sdai2
raw /dev/raw/raw115 /dev/sdaj2
raw /dev/raw/raw116 /dev/sdak2
raw /dev/raw/raw117 /dev/sdal2
raw /dev/raw/raw118 /dev/sdam2
raw /dev/raw/raw119 /dev/sdan2
raw /dev/raw/raw120 /dev/sdao2
raw /dev/raw/raw121 /dev/sdap2
raw /dev/raw/raw122 /dev/sdaq2
raw /dev/raw/raw123 /dev/sdar2
raw /dev/raw/raw124 /dev/sdas2
raw /dev/raw/raw125 /dev/sdat2
raw /dev/raw/raw126 /dev/sdau2
raw /dev/raw/raw127 /dev/sdav2
raw /dev/raw/raw128 /dev/sdaw2
raw /dev/raw/raw129 /dev/sdax2
raw /dev/raw/raw130 /dev/sday2
raw /dev/raw/raw131 /dev/sdaz2
raw /dev/raw/raw132 /dev/sdba2
raw /dev/raw/raw133 /dev/sdbb2
raw /dev/raw/raw134 /dev/sdbc2
raw /dev/raw/raw135 /dev/sdbd2
raw /dev/raw/raw136 /dev/sdbe2
raw /dev/raw/raw137 /dev/sdbf2
raw /dev/raw/raw138 /dev/sdbg2
raw /dev/raw/raw139 /dev/sdbh2
raw /dev/raw/raw140 /dev/sdbi2
raw /dev/raw/raw141 /dev/sdbj2
raw /dev/raw/raw142 /dev/sdbk2
raw /dev/raw/raw143 /dev/sdbl2
raw /dev/raw/raw144 /dev/sdbm2
raw /dev/raw/raw145 /dev/sdbn2
raw /dev/raw/raw146 /dev/sdbo2
raw /dev/raw/raw147 /dev/sdbp2
raw /dev/raw/raw148 /dev/sdbq2
raw /dev/raw/raw149 /dev/sdbr2
raw /dev/raw/raw150 /dev/sdbs2
raw /dev/raw/raw151 /dev/sdbt2
raw /dev/raw/raw152 /dev/sdbu2
raw /dev/raw/raw153 /dev/sdbv2
raw /dev/raw/raw154 /dev/sdbw2
raw /dev/raw/raw155 /dev/sdbx2
raw /dev/raw/raw156 /dev/sdby2
raw /dev/raw/raw157 /dev/sdbz2
raw /dev/raw/raw158 /dev/sdca2
raw /dev/raw/raw159 /dev/sdcb2
raw /dev/raw/raw160 /dev/sdcc2

raw /dev/raw/raw161 /dev/sdb3
raw /dev/raw/raw162 /dev/sdc3
raw /dev/raw/raw163 /dev/sdd3
raw /dev/raw/raw164 /dev/sde3
raw /dev/raw/raw165 /dev/sdf3
raw /dev/raw/raw166 /dev/sdg3
raw /dev/raw/raw167 /dev/sdh3
raw /dev/raw/raw168 /dev/sdi3
raw /dev/raw/raw169 /dev/sdj3
raw /dev/raw/raw170 /dev/sdk3
raw /dev/raw/raw171 /dev/sdl3
raw /dev/raw/raw172 /dev/sdm3
raw /dev/raw/raw173 /dev/sdn3
raw /dev/raw/raw174 /dev/sdo3
raw /dev/raw/raw175 /dev/sdp3
raw /dev/raw/raw176 /dev/sdq3
raw /dev/raw/raw177 /dev/sdr3
raw /dev/raw/raw178 /dev/sds3
raw /dev/raw/raw179 /dev/sdt3
raw /dev/raw/raw180 /dev/sdu3
raw /dev/raw/raw181 /dev/sdv3
raw /dev/raw/raw182 /dev/sdw3
raw /dev/raw/raw183 /dev/sdx3
raw /dev/raw/raw184 /dev/sdy3
raw /dev/raw/raw185 /dev/sdz3
raw /dev/raw/raw186 /dev/sdaa3
raw /dev/raw/raw187 /dev/sdab3
raw /dev/raw/raw188 /dev/sdac3
raw /dev/raw/raw189 /dev/sdad3
raw /dev/raw/raw190 /dev/sdae3
raw /dev/raw/raw191 /dev/sdaf3
raw /dev/raw/raw192 /dev/sdag3
raw /dev/raw/raw193 /dev/sdah3
raw /dev/raw/raw194 /dev/sdai3
raw /dev/raw/raw195 /dev/sdaj3
raw /dev/raw/raw196 /dev/sdak3
raw /dev/raw/raw197 /dev/sdal3
raw /dev/raw/raw198 /dev/sdam3
raw /dev/raw/raw199 /dev/sdan3
raw /dev/raw/raw200 /dev/sdao3
raw /dev/raw/raw201 /dev/sdap3
raw /dev/raw/raw202 /dev/sdaq3
raw /dev/raw/raw203 /dev/sdar3
raw /dev/raw/raw204 /dev/sdas3
raw /dev/raw/raw205 /dev/sdat3
raw /dev/raw/raw206 /dev/sdau3
raw /dev/raw/raw207 /dev/sdav3
raw /dev/raw/raw208 /dev/sdaw3
raw /dev/raw/raw209 /dev/sdax3
raw /dev/raw/raw210 /dev/sday3
raw /dev/raw/raw211 /dev/sdaz3
raw /dev/raw/raw212 /dev/sdba3
raw /dev/raw/raw213 /dev/sdbb3
raw /dev/raw/raw214 /dev/sdbc3
raw /dev/raw/raw215 /dev/sdbd3
raw /dev/raw/raw216 /dev/sdbe3
raw /dev/raw/raw217 /dev/sdbf3
raw /dev/raw/raw218 /dev/sdbg3
raw /dev/raw/raw219 /dev/sdbh3
raw /dev/raw/raw220 /dev/sdbi3
raw /dev/raw/raw221 /dev/sdbj3
raw /dev/raw/raw222 /dev/sdbk3
raw /dev/raw/raw223 /dev/sdbl3
raw /dev/raw/raw224 /dev/sdbm3
raw /dev/raw/raw225 /dev/sdbn3
raw /dev/raw/raw226 /dev/sdbo3
raw /dev/raw/raw227 /dev/sdbp3
raw /dev/raw/raw228 /dev/sdbq3
raw /dev/raw/raw229 /dev/sdbr3
raw /dev/raw/raw230 /dev/sdbs3
raw /dev/raw/raw231 /dev/sdbt3
raw /dev/raw/raw232 /dev/sdbu3
raw /dev/raw/raw233 /dev/sdbv3
raw /dev/raw/raw234 /dev/sdbw3
raw /dev/raw/raw235 /dev/sdbx3
raw /dev/raw/raw236 /dev/sdby3
raw /dev/raw/raw237 /dev/sdbz3
raw /dev/raw/raw238 /dev/sdca3
raw /dev/raw/raw239 /dev/sdcb3
raw /dev/raw/raw240 /dev/sdcc3

raw /dev/raw/raw241 /dev/sdb5
raw /dev/raw/raw242 /dev/sdc5
raw /dev/raw/raw243 /dev/sdd5
raw /dev/raw/raw244 /dev/sde5
raw /dev/raw/raw245 /dev/sdf5
raw /dev/raw/raw246 /dev/sdg5
raw /dev/raw/raw247 /dev/sdh5
raw /dev/raw/raw248 /dev/sdi5
raw /dev/raw/raw249 /dev/sdj5
raw /dev/raw/raw250 /dev/sdk5
raw /dev/raw/raw251 /dev/sdl5
raw /dev/raw/raw252 /dev/sdm5
raw /dev/raw/raw253 /dev/sdn5
raw /dev/raw/raw254 /dev/sdo5
raw /dev/raw/raw255 /dev/sdp5
raw /dev/raw/raw256 /dev/sdq5
raw /dev/raw/raw257 /dev/sdr5
raw /dev/raw/raw258 /dev/sds5
raw /dev/raw/raw259 /dev/sdt5
raw /dev/raw/raw260 /dev/sdu5
raw /dev/raw/raw261 /dev/sdv5
raw /dev/raw/raw262 /dev/sdw5
raw /dev/raw/raw263 /dev/sdx5
raw /dev/raw/raw264 /dev/sdy5
raw /dev/raw/raw265 /dev/sdz5
raw /dev/raw/raw266 /dev/sdaa5
raw /dev/raw/raw267 /dev/sdab5
raw /dev/raw/raw268 /dev/sdac5
raw /dev/raw/raw269 /dev/sdad5
raw /dev/raw/raw270 /dev/sdae5
raw /dev/raw/raw271 /dev/sdaf5
raw /dev/raw/raw272 /dev/sdag5
raw /dev/raw/raw273 /dev/sdah5
raw /dev/raw/raw274 /dev/sdai5
raw /dev/raw/raw275 /dev/sdaj5
raw /dev/raw/raw276 /dev/sdak5
raw /dev/raw/raw277 /dev/sdal5
raw /dev/raw/raw278 /dev/sdam5
raw /dev/raw/raw279 /dev/sdan5
raw /dev/raw/raw280 /dev/sdao5
raw /dev/raw/raw281 /dev/sdap5
raw /dev/raw/raw282 /dev/sdaq5
raw /dev/raw/raw283 /dev/sdar5
raw /dev/raw/raw284 /dev/sdas5
raw /dev/raw/raw285 /dev/sdat5
raw /dev/raw/raw286 /dev/sdau5
raw /dev/raw/raw287 /dev/sdav5
raw /dev/raw/raw288 /dev/sdaw5
raw /dev/raw/raw289 /dev/sdax5
raw /dev/raw/raw290 /dev/sday5
raw /dev/raw/raw291 /dev/sdaz5
raw /dev/raw/raw292 /dev/sdba5
raw /dev/raw/raw293 /dev/sdbb5
raw /dev/raw/raw294 /dev/sdbc5
raw /dev/raw/raw295 /dev/sdbd5
raw /dev/raw/raw296 /dev/sdbe5
raw /dev/raw/raw297 /dev/sdbf5

```
raw /dev/raw/raw298    /dev/sdbg5
raw /dev/raw/raw299    /dev/sdbh5
raw /dev/raw/raw300    /dev/sdbi5
raw /dev/raw/raw301    /dev/sdbj5
raw /dev/raw/raw302    /dev/sdbk5
raw /dev/raw/raw303    /dev/sdbl5
raw /dev/raw/raw304    /dev/sdbm5
raw /dev/raw/raw305    /dev/sdbn5
raw /dev/raw/raw306    /dev/sdbo5
raw /dev/raw/raw307    /dev/sdbp5
raw /dev/raw/raw308    /dev/sdbq5
raw /dev/raw/raw309    /dev/sdbr5
raw /dev/raw/raw310    /dev/sdbs5
raw /dev/raw/raw311    /dev/sdbt5
raw /dev/raw/raw312    /dev/sdbu5
raw /dev/raw/raw313    /dev/sdbv5
raw /dev/raw/raw314    /dev/sdbw5
raw /dev/raw/raw315    /dev/sdbx5
raw /dev/raw/raw316    /dev/sdby5
raw /dev/raw/raw317    /dev/sdbz5
raw /dev/raw/raw318    /dev/sdca5
raw /dev/raw/raw319    /dev/sdcb5
raw /dev/raw/raw320    /dev/sdcc5

raw /dev/raw/raw321    /dev/sdb6
raw /dev/raw/raw322    /dev/sdc6
raw /dev/raw/raw323    /dev/sdd6
raw /dev/raw/raw324    /dev/sde6
raw /dev/raw/raw325    /dev/sdf6
raw /dev/raw/raw326    /dev/sdg6
raw /dev/raw/raw327    /dev/sdh6
raw /dev/raw/raw328    /dev/sdi6
raw /dev/raw/raw329    /dev/sdj6
raw /dev/raw/raw330    /dev/sdk6
raw /dev/raw/raw331    /dev/sdl6
raw /dev/raw/raw332    /dev/sdm6
raw /dev/raw/raw333    /dev/sdn6
raw /dev/raw/raw334    /dev/sdo6
raw /dev/raw/raw335    /dev/sdp6
raw /dev/raw/raw336    /dev/sdq6
raw /dev/raw/raw337    /dev/sdr6
raw /dev/raw/raw338    /dev/sds6
raw /dev/raw/raw339    /dev/sdt6
raw /dev/raw/raw340    /dev/sdu6
raw /dev/raw/raw341    /dev/sdv6
raw /dev/raw/raw342    /dev/sdw6
raw /dev/raw/raw343    /dev/sdx6
raw /dev/raw/raw344    /dev/sdy6
raw /dev/raw/raw345    /dev/sdz6
raw /dev/raw/raw346    /dev/sdaa6
raw /dev/raw/raw347    /dev/sdab6
raw /dev/raw/raw348    /dev/sdac6
raw /dev/raw/raw349    /dev/sdad6
raw /dev/raw/raw350    /dev/sdae6
raw /dev/raw/raw351    /dev/sdaf6
raw /dev/raw/raw352    /dev/sdag6
raw /dev/raw/raw353    /dev/sdah6
raw /dev/raw/raw354    /dev/sdai6
raw /dev/raw/raw355    /dev/sdaj6
raw /dev/raw/raw356    /dev/sdak6
raw /dev/raw/raw357    /dev/sdal6
raw /dev/raw/raw358    /dev/sdam6
raw /dev/raw/raw359    /dev/sdan6
raw /dev/raw/raw360    /dev/sdao6
raw /dev/raw/raw361    /dev/sdap6
raw /dev/raw/raw362    /dev/sdaq6
raw /dev/raw/raw363    /dev/sdar6
raw /dev/raw/raw364    /dev/sdas6
raw /dev/raw/raw365    /dev/sdat6
raw /dev/raw/raw366    /dev/sdau6
raw /dev/raw/raw367    /dev/sdav6
raw /dev/raw/raw368    /dev/sdaw6
raw /dev/raw/raw369    /dev/sdax6
raw /dev/raw/raw370    /dev/sday6
raw /dev/raw/raw371    /dev/sdaz6
raw /dev/raw/raw372    /dev/sdba6
raw /dev/raw/raw373    /dev/sdbb6
raw /dev/raw/raw374    /dev/sdbc6
raw /dev/raw/raw375    /dev/sdbd6
raw /dev/raw/raw376    /dev/sdbe6
raw /dev/raw/raw377    /dev/sdbf6
raw /dev/raw/raw378    /dev/sdbg6
raw /dev/raw/raw379    /dev/sdbh6
raw /dev/raw/raw380    /dev/sdbi6
raw /dev/raw/raw381    /dev/sdbj6
raw /dev/raw/raw382    /dev/sdbk6
raw /dev/raw/raw383    /dev/sdbl6
raw /dev/raw/raw384    /dev/sdbm6
raw /dev/raw/raw385    /dev/sdbn6
raw /dev/raw/raw386    /dev/sdbo6
raw /dev/raw/raw387    /dev/sdbp6
raw /dev/raw/raw388    /dev/sdbq6
raw /dev/raw/raw389    /dev/sdbr6
raw /dev/raw/raw390    /dev/sdbs6
raw /dev/raw/raw391    /dev/sdbt6
raw /dev/raw/raw392    /dev/sdbu6
raw /dev/raw/raw393    /dev/sdbv6
raw /dev/raw/raw394    /dev/sdbw6
raw /dev/raw/raw395    /dev/sdbx6
raw /dev/raw/raw396    /dev/sdby6
raw /dev/raw/raw397    /dev/sdbz6
raw /dev/raw/raw398    /dev/sdca6
raw /dev/raw/raw399    /dev/sdcb6
raw /dev/raw/raw400    /dev/sdcc6

raw /dev/raw/raw401    /dev/sdb7
raw /dev/raw/raw402    /dev/sdc7
raw /dev/raw/raw403    /dev/sdd7
raw /dev/raw/raw404    /dev/sde7
raw /dev/raw/raw405    /dev/sdf7
raw /dev/raw/raw406    /dev/sdg7
raw /dev/raw/raw407    /dev/sdh7
raw /dev/raw/raw408    /dev/sdi7
raw /dev/raw/raw409    /dev/sdj7
raw /dev/raw/raw410    /dev/sdk7
raw /dev/raw/raw411    /dev/sdl7
raw /dev/raw/raw412    /dev/sdm7
raw /dev/raw/raw413    /dev/sdn7
raw /dev/raw/raw414    /dev/sdo7
raw /dev/raw/raw415    /dev/sdp7
raw /dev/raw/raw416    /dev/sdq7
raw /dev/raw/raw417    /dev/sdr7
raw /dev/raw/raw418    /dev/sds7
raw /dev/raw/raw419    /dev/sdt7
raw /dev/raw/raw420    /dev/sdu7
raw /dev/raw/raw421    /dev/sdv7
raw /dev/raw/raw422    /dev/sdw7
raw /dev/raw/raw423    /dev/sdx7
raw /dev/raw/raw424    /dev/sdy7
raw /dev/raw/raw425    /dev/sdz7
raw /dev/raw/raw426    /dev/sdaa7
raw /dev/raw/raw427    /dev/sdab7
raw /dev/raw/raw428    /dev/sdac7
raw /dev/raw/raw429    /dev/sdad7
raw /dev/raw/raw430    /dev/sdae7
raw /dev/raw/raw431    /dev/sdaf7
raw /dev/raw/raw432    /dev/sdag7
raw /dev/raw/raw433    /dev/sdah7
raw /dev/raw/raw434    /dev/sdai7
raw /dev/raw/raw435    /dev/sdaj7
raw /dev/raw/raw436    /dev/sdak7
raw /dev/raw/raw437    /dev/sdal7
raw /dev/raw/raw438    /dev/sdam7
raw /dev/raw/raw439    /dev/sdan7
raw /dev/raw/raw440    /dev/sdao7
raw /dev/raw/raw441    /dev/sdap7
raw /dev/raw/raw442    /dev/sdaq7
raw /dev/raw/raw443    /dev/sdar7
raw /dev/raw/raw444    /dev/sdas7
raw /dev/raw/raw445    /dev/sdat7
raw /dev/raw/raw446    /dev/sdau7
raw /dev/raw/raw447    /dev/sdav7
raw /dev/raw/raw448    /dev/sdaw7
raw /dev/raw/raw449    /dev/sdax7
raw /dev/raw/raw450    /dev/sday7
raw /dev/raw/raw451    /dev/sdaz7
raw /dev/raw/raw452    /dev/sdba7
raw /dev/raw/raw453    /dev/sdbb7
raw /dev/raw/raw454    /dev/sdbc7
raw /dev/raw/raw455    /dev/sdbd7
raw /dev/raw/raw456    /dev/sdbe7
raw /dev/raw/raw457    /dev/sdbf7
raw /dev/raw/raw458    /dev/sdbg7
raw /dev/raw/raw459    /dev/sdbh7
raw /dev/raw/raw460    /dev/sdbi7
raw /dev/raw/raw461    /dev/sdbj7
raw /dev/raw/raw462    /dev/sdbk7
raw /dev/raw/raw463    /dev/sdbl7
raw /dev/raw/raw464    /dev/sdbm7
raw /dev/raw/raw465    /dev/sdbn7
raw /dev/raw/raw466    /dev/sdbo7
raw /dev/raw/raw467    /dev/sdbp7
raw /dev/raw/raw468    /dev/sdbq7
raw /dev/raw/raw469    /dev/sdbr7
raw /dev/raw/raw470    /dev/sdbs7
raw /dev/raw/raw471    /dev/sdbt7
raw /dev/raw/raw472    /dev/sdbu7
raw /dev/raw/raw473    /dev/sdbv7
raw /dev/raw/raw474    /dev/sdbw7
raw /dev/raw/raw475    /dev/sdbx7
raw /dev/raw/raw476    /dev/sdby7
raw /dev/raw/raw477    /dev/sdbz7
raw /dev/raw/raw478    /dev/sdca7
raw /dev/raw/raw479    /dev/sdcb7
raw /dev/raw/raw480    /dev/sdcc7

raw /dev/raw/raw481    /dev/sdb8
raw /dev/raw/raw482    /dev/sdc8
raw /dev/raw/raw483    /dev/sdd8
raw /dev/raw/raw484    /dev/sde8
raw /dev/raw/raw485    /dev/sdf8
raw /dev/raw/raw486    /dev/sdg8
raw /dev/raw/raw487    /dev/sdh8
raw /dev/raw/raw488    /dev/sdi8
raw /dev/raw/raw489    /dev/sdj8
raw /dev/raw/raw490    /dev/sdk8
raw /dev/raw/raw491    /dev/sdl8
raw /dev/raw/raw492    /dev/sdm8
raw /dev/raw/raw493    /dev/sdn8
raw /dev/raw/raw494    /dev/sdo8
raw /dev/raw/raw495    /dev/sdp8
raw /dev/raw/raw496    /dev/sdq8
raw /dev/raw/raw497    /dev/sdr8
raw /dev/raw/raw498    /dev/sds8
raw /dev/raw/raw499    /dev/sdt8
raw /dev/raw/raw500    /dev/sdu8
raw /dev/raw/raw501    /dev/sdv8
raw /dev/raw/raw502    /dev/sdw8
raw /dev/raw/raw503    /dev/sdx8
raw /dev/raw/raw504    /dev/sdy8
```

```
raw /dev/raw/raw505   /dev/sdz8
raw /dev/raw/raw506   /dev/sdaa8
raw /dev/raw/raw507   /dev/sdab8
raw /dev/raw/raw508   /dev/sdac8
raw /dev/raw/raw509   /dev/sdad8
raw /dev/raw/raw510   /dev/sdae8
raw /dev/raw/raw511   /dev/sdaf8
raw /dev/raw/raw512   /dev/sdag8
raw /dev/raw/raw513   /dev/sdah8
raw /dev/raw/raw514   /dev/sdai8
raw /dev/raw/raw515   /dev/sdaj8
raw /dev/raw/raw516   /dev/sdak8
raw /dev/raw/raw517   /dev/sdal8
raw /dev/raw/raw518   /dev/sdam8
raw /dev/raw/raw519   /dev/sdan8
raw /dev/raw/raw520   /dev/sdao8
raw /dev/raw/raw521   /dev/sdap8
raw /dev/raw/raw522   /dev/sdaq8
raw /dev/raw/raw523   /dev/sdar8
raw /dev/raw/raw524   /dev/sdas8
raw /dev/raw/raw525   /dev/sdat8
raw /dev/raw/raw526   /dev/sdau8
raw /dev/raw/raw527   /dev/sdav8
raw /dev/raw/raw528   /dev/sdaw8
raw /dev/raw/raw529   /dev/sdax8
raw /dev/raw/raw530   /dev/sday8
raw /dev/raw/raw531   /dev/sdaz8
raw /dev/raw/raw532   /dev/sdba8
raw /dev/raw/raw533   /dev/sdbb8
raw /dev/raw/raw534   /dev/sdbc8
raw /dev/raw/raw535   /dev/sdbd8
raw /dev/raw/raw536   /dev/sdbe8
raw /dev/raw/raw537   /dev/sdbf8
raw /dev/raw/raw538   /dev/sdbg8
raw /dev/raw/raw539   /dev/sdbh8
raw /dev/raw/raw540   /dev/sdbi8
raw /dev/raw/raw541   /dev/sdbj8
raw /dev/raw/raw542   /dev/sdbk8
raw /dev/raw/raw543   /dev/sdbl8
raw /dev/raw/raw544   /dev/sdbm8
raw /dev/raw/raw545   /dev/sdbn8
raw /dev/raw/raw546   /dev/sdbo8
raw /dev/raw/raw547   /dev/sdbp8
raw /dev/raw/raw548   /dev/sdbq8
raw /dev/raw/raw549   /dev/sdbr8
raw /dev/raw/raw550   /dev/sdbs8
raw /dev/raw/raw551   /dev/sdbt8
raw /dev/raw/raw552   /dev/sdbu8
raw /dev/raw/raw553   /dev/sdbv8
raw /dev/raw/raw554   /dev/sdbw8
raw /dev/raw/raw555   /dev/sdbx8
raw /dev/raw/raw556   /dev/sdby8
raw /dev/raw/raw557   /dev/sdbz8
raw /dev/raw/raw558   /dev/sdca8
raw /dev/raw/raw559   /dev/sdcb8
raw /dev/raw/raw560   /dev/sdcc8

raw /dev/raw/raw561   /dev/sdb9
raw /dev/raw/raw562   /dev/sdc9
raw /dev/raw/raw563   /dev/sdd9
raw /dev/raw/raw564   /dev/sde9
raw /dev/raw/raw565   /dev/sdf9
raw /dev/raw/raw566   /dev/sdg9
raw /dev/raw/raw567   /dev/sdh9
raw /dev/raw/raw568   /dev/sdi9
raw /dev/raw/raw569   /dev/sdj9
raw /dev/raw/raw570   /dev/sdk9
raw /dev/raw/raw571   /dev/sdl9
raw /dev/raw/raw572   /dev/sdm9
raw /dev/raw/raw573   /dev/sdn9
raw /dev/raw/raw574   /dev/sdo9
raw /dev/raw/raw575   /dev/sdp9
raw /dev/raw/raw576   /dev/sdq9
raw /dev/raw/raw577   /dev/sdr9
raw /dev/raw/raw578   /dev/sds9
raw /dev/raw/raw579   /dev/sdt9
raw /dev/raw/raw580   /dev/sdu9
raw /dev/raw/raw581   /dev/sdv9
raw /dev/raw/raw582   /dev/sdw9
raw /dev/raw/raw583   /dev/sdx9
raw /dev/raw/raw584   /dev/sdy9
raw /dev/raw/raw585   /dev/sdz9
raw /dev/raw/raw586   /dev/sdaa9
raw /dev/raw/raw587   /dev/sdab9
raw /dev/raw/raw588   /dev/sdac9
raw /dev/raw/raw589   /dev/sdad9
raw /dev/raw/raw590   /dev/sdae9
raw /dev/raw/raw591   /dev/sdaf9
raw /dev/raw/raw592   /dev/sdag9
raw /dev/raw/raw593   /dev/sdah9
raw /dev/raw/raw594   /dev/sdai9
raw /dev/raw/raw595   /dev/sdaj9
raw /dev/raw/raw596   /dev/sdak9
raw /dev/raw/raw597   /dev/sdal9
raw /dev/raw/raw598   /dev/sdam9
raw /dev/raw/raw599   /dev/sdan9
raw /dev/raw/raw600   /dev/sdao9
raw /dev/raw/raw601   /dev/sdap9
raw /dev/raw/raw602   /dev/sdaq9
raw /dev/raw/raw603   /dev/sdar9
raw /dev/raw/raw604   /dev/sdas9
raw /dev/raw/raw605   /dev/sdat9
raw /dev/raw/raw606   /dev/sdau9
raw /dev/raw/raw607   /dev/sdav9
raw /dev/raw/raw608   /dev/sdaw9
raw /dev/raw/raw609   /dev/sdax9
raw /dev/raw/raw610   /dev/sday9
raw /dev/raw/raw611   /dev/sdaz9
raw /dev/raw/raw612   /dev/sdba9
raw /dev/raw/raw613   /dev/sdbb9
raw /dev/raw/raw614   /dev/sdbc9
raw /dev/raw/raw615   /dev/sdbd9
raw /dev/raw/raw616   /dev/sdbe9
raw /dev/raw/raw617   /dev/sdbf9
raw /dev/raw/raw618   /dev/sdbg9
raw /dev/raw/raw619   /dev/sdbh9
raw /dev/raw/raw620   /dev/sdbi9
raw /dev/raw/raw621   /dev/sdbj9
raw /dev/raw/raw622   /dev/sdbk9
raw /dev/raw/raw623   /dev/sdbl9
raw /dev/raw/raw624   /dev/sdbm9
raw /dev/raw/raw625   /dev/sdbn9
raw /dev/raw/raw626   /dev/sdbo9
raw /dev/raw/raw627   /dev/sdbp9
raw /dev/raw/raw628   /dev/sdbq9
raw /dev/raw/raw629   /dev/sdbr9
raw /dev/raw/raw630   /dev/sdbs9
raw /dev/raw/raw631   /dev/sdbt9
raw /dev/raw/raw632   /dev/sdbu9
raw /dev/raw/raw633   /dev/sdbv9
raw /dev/raw/raw634   /dev/sdbw9
raw /dev/raw/raw635   /dev/sdbx9
raw /dev/raw/raw636   /dev/sdby9
raw /dev/raw/raw637   /dev/sdbz9
raw /dev/raw/raw638   /dev/sdca9
raw /dev/raw/raw639   /dev/sdcb9
raw /dev/raw/raw640   /dev/sdcc9

raw /dev/raw/raw641   /dev/sdb10
raw /dev/raw/raw642   /dev/sdc10
raw /dev/raw/raw643   /dev/sdd10
raw /dev/raw/raw644   /dev/sde10
raw /dev/raw/raw645   /dev/sdf10
raw /dev/raw/raw646   /dev/sdg10
raw /dev/raw/raw647   /dev/sdh10
raw /dev/raw/raw648   /dev/sdi10
raw /dev/raw/raw649   /dev/sdj10
raw /dev/raw/raw650   /dev/sdk10
raw /dev/raw/raw651   /dev/sdl10
raw /dev/raw/raw652   /dev/sdm10
raw /dev/raw/raw653   /dev/sdn10
raw /dev/raw/raw654   /dev/sdo10
raw /dev/raw/raw655   /dev/sdp10
raw /dev/raw/raw656   /dev/sdq10
raw /dev/raw/raw657   /dev/sdr10
raw /dev/raw/raw658   /dev/sds10
raw /dev/raw/raw659   /dev/sdt10
raw /dev/raw/raw660   /dev/sdu10
raw /dev/raw/raw661   /dev/sdv10
raw /dev/raw/raw662   /dev/sdw10
raw /dev/raw/raw663   /dev/sdx10
raw /dev/raw/raw664   /dev/sdy10
raw /dev/raw/raw665   /dev/sdz10
raw /dev/raw/raw666   /dev/sdaa10
raw /dev/raw/raw667   /dev/sdab10
raw /dev/raw/raw668   /dev/sdac10
raw /dev/raw/raw669   /dev/sdad10
raw /dev/raw/raw670   /dev/sdae10
raw /dev/raw/raw671   /dev/sdaf10
raw /dev/raw/raw672   /dev/sdag10
raw /dev/raw/raw673   /dev/sdah10
raw /dev/raw/raw674   /dev/sdai10
raw /dev/raw/raw675   /dev/sdaj10
raw /dev/raw/raw676   /dev/sdak10
raw /dev/raw/raw677   /dev/sdal10
raw /dev/raw/raw678   /dev/sdam10
raw /dev/raw/raw679   /dev/sdan10
raw /dev/raw/raw680   /dev/sdao10
raw /dev/raw/raw681   /dev/sdap10
raw /dev/raw/raw682   /dev/sdaq10
raw /dev/raw/raw683   /dev/sdar10
raw /dev/raw/raw684   /dev/sdas10
raw /dev/raw/raw685   /dev/sdat10
raw /dev/raw/raw686   /dev/sdau10
raw /dev/raw/raw687   /dev/sdav10
raw /dev/raw/raw688   /dev/sdaw10
raw /dev/raw/raw689   /dev/sdax10
raw /dev/raw/raw690   /dev/sday10
raw /dev/raw/raw691   /dev/sdaz10
raw /dev/raw/raw692   /dev/sdba10
raw /dev/raw/raw693   /dev/sdbb10
raw /dev/raw/raw694   /dev/sdbc10
raw /dev/raw/raw695   /dev/sdbd10
raw /dev/raw/raw696   /dev/sdbe10
raw /dev/raw/raw697   /dev/sdbf10
raw /dev/raw/raw698   /dev/sdbg10
raw /dev/raw/raw699   /dev/sdbh10
raw /dev/raw/raw700   /dev/sdbi10
raw /dev/raw/raw701   /dev/sdbj10
raw /dev/raw/raw702   /dev/sdbk10
raw /dev/raw/raw703   /dev/sdbl10
raw /dev/raw/raw704   /dev/sdbm10
raw /dev/raw/raw705   /dev/sdbn10
raw /dev/raw/raw706   /dev/sdbo10
raw /dev/raw/raw707   /dev/sdbp10
raw /dev/raw/raw708   /dev/sdbq10
raw /dev/raw/raw709   /dev/sdbr10
raw /dev/raw/raw710   /dev/sdbs10
raw /dev/raw/raw711   /dev/sdbt10
raw /dev/raw/raw712   /dev/sdbu10
```

```
raw /dev/raw/raw713  /dev/sdbv10
raw /dev/raw/raw714  /dev/sdbw10
raw /dev/raw/raw715  /dev/sdbx10
raw /dev/raw/raw716  /dev/sdby10
raw /dev/raw/raw717  /dev/sdbz10
raw /dev/raw/raw718  /dev/sdca10
raw /dev/raw/raw719  /dev/sdcb10
raw /dev/raw/raw720  /dev/sdcc10

raw /dev/raw/raw721  /dev/sdb11
raw /dev/raw/raw722  /dev/sdc11
raw /dev/raw/raw723  /dev/sdd11
raw /dev/raw/raw724  /dev/sde11
raw /dev/raw/raw725  /dev/sdf11
raw /dev/raw/raw726  /dev/sdg11
raw /dev/raw/raw727  /dev/sdh11
raw /dev/raw/raw728  /dev/sdi11
raw /dev/raw/raw729  /dev/sdj11
raw /dev/raw/raw730  /dev/sdk11
raw /dev/raw/raw731  /dev/sdl11
raw /dev/raw/raw732  /dev/sdm11
raw /dev/raw/raw733  /dev/sdn11
raw /dev/raw/raw734  /dev/sdo11
raw /dev/raw/raw735  /dev/sdp11
raw /dev/raw/raw736  /dev/sdq11
raw /dev/raw/raw737  /dev/sdr11
raw /dev/raw/raw738  /dev/sds11
raw /dev/raw/raw739  /dev/sdt11
raw /dev/raw/raw740  /dev/sdu11
raw /dev/raw/raw741  /dev/sdv11
raw /dev/raw/raw742  /dev/sdw11
raw /dev/raw/raw743  /dev/sdx11
raw /dev/raw/raw744  /dev/sdy11
raw /dev/raw/raw745  /dev/sdz11
raw /dev/raw/raw746  /dev/sdaa11
raw /dev/raw/raw747  /dev/sdab11
raw /dev/raw/raw748  /dev/sdac11
raw /dev/raw/raw749  /dev/sdad11
raw /dev/raw/raw750  /dev/sdae11
raw /dev/raw/raw751  /dev/sdaf11
raw /dev/raw/raw752  /dev/sdag11
raw /dev/raw/raw753  /dev/sdah11
raw /dev/raw/raw754  /dev/sdai11
raw /dev/raw/raw755  /dev/sdaj11
raw /dev/raw/raw756  /dev/sdak11
raw /dev/raw/raw757  /dev/sdal11
raw /dev/raw/raw758  /dev/sdam11
raw /dev/raw/raw759  /dev/sdan11
raw /dev/raw/raw760  /dev/sdao11
raw /dev/raw/raw761  /dev/sdap11
raw /dev/raw/raw762  /dev/sdaq11
raw /dev/raw/raw763  /dev/sdar11
raw /dev/raw/raw764  /dev/sdas11
raw /dev/raw/raw765  /dev/sdat11
raw /dev/raw/raw766  /dev/sdau11
raw /dev/raw/raw767  /dev/sdav11
raw /dev/raw/raw768  /dev/sdaw11
raw /dev/raw/raw769  /dev/sdax11
raw /dev/raw/raw770  /dev/sday11
raw /dev/raw/raw771  /dev/sdaz11
raw /dev/raw/raw772  /dev/sdba11
raw /dev/raw/raw773  /dev/sdbb11
raw /dev/raw/raw774  /dev/sdbc11
raw /dev/raw/raw775  /dev/sdbd11
raw /dev/raw/raw776  /dev/sdbe11
raw /dev/raw/raw777  /dev/sdbf11
raw /dev/raw/raw778  /dev/sdbg11
raw /dev/raw/raw779  /dev/sdbh11
raw /dev/raw/raw780  /dev/sdbi11
raw /dev/raw/raw781  /dev/sdbj11
raw /dev/raw/raw782  /dev/sdbk11
raw /dev/raw/raw783  /dev/sdbl11
raw /dev/raw/raw784  /dev/sdbm11
raw /dev/raw/raw785  /dev/sdbn11
raw /dev/raw/raw786  /dev/sdbo11
raw /dev/raw/raw787  /dev/sdbp11
raw /dev/raw/raw788  /dev/sdbq11
raw /dev/raw/raw789  /dev/sdbr11
raw /dev/raw/raw790  /dev/sdbs11
raw /dev/raw/raw791  /dev/sdbt11
raw /dev/raw/raw792  /dev/sdbu11
raw /dev/raw/raw793  /dev/sdbv11
raw /dev/raw/raw794  /dev/sdbw11
raw /dev/raw/raw795  /dev/sdbx11
raw /dev/raw/raw796  /dev/sdby11
raw /dev/raw/raw797  /dev/sdbz11
raw /dev/raw/raw798  /dev/sdca11
raw /dev/raw/raw799  /dev/sdcb11
raw /dev/raw/raw800  /dev/sdcc11

raw /dev/raw/raw801  /dev/sdb12
raw /dev/raw/raw802  /dev/sdc12
raw /dev/raw/raw803  /dev/sdd12
raw /dev/raw/raw804  /dev/sde12
raw /dev/raw/raw805  /dev/sdf12
raw /dev/raw/raw806  /dev/sdg12
raw /dev/raw/raw807  /dev/sdh12
raw /dev/raw/raw808  /dev/sdi12
raw /dev/raw/raw809  /dev/sdj12
raw /dev/raw/raw810  /dev/sdk12
raw /dev/raw/raw811  /dev/sdl12
raw /dev/raw/raw812  /dev/sdm12
raw /dev/raw/raw813  /dev/sdn12
raw /dev/raw/raw814  /dev/sdo12
raw /dev/raw/raw815  /dev/sdp12
raw /dev/raw/raw816  /dev/sdq12
raw /dev/raw/raw817  /dev/sdr12
raw /dev/raw/raw818  /dev/sds12
raw /dev/raw/raw819  /dev/sdt12
raw /dev/raw/raw820  /dev/sdu12
raw /dev/raw/raw821  /dev/sdv12
raw /dev/raw/raw822  /dev/sdw12
raw /dev/raw/raw823  /dev/sdx12
raw /dev/raw/raw824  /dev/sdy12
raw /dev/raw/raw825  /dev/sdz12
raw /dev/raw/raw826  /dev/sdaa12
raw /dev/raw/raw827  /dev/sdab12
raw /dev/raw/raw828  /dev/sdac12
raw /dev/raw/raw829  /dev/sdad12
raw /dev/raw/raw830  /dev/sdae12
raw /dev/raw/raw831  /dev/sdaf12
raw /dev/raw/raw832  /dev/sdag12
raw /dev/raw/raw833  /dev/sdah12
raw /dev/raw/raw834  /dev/sdai12
raw /dev/raw/raw835  /dev/sdaj12
raw /dev/raw/raw836  /dev/sdak12
raw /dev/raw/raw837  /dev/sdal12
raw /dev/raw/raw838  /dev/sdam12
raw /dev/raw/raw839  /dev/sdan12
raw /dev/raw/raw840  /dev/sdao12
raw /dev/raw/raw841  /dev/sdap12
raw /dev/raw/raw842  /dev/sdaq12
raw /dev/raw/raw843  /dev/sdar12
raw /dev/raw/raw844  /dev/sdas12
raw /dev/raw/raw845  /dev/sdat12
raw /dev/raw/raw846  /dev/sdau12
raw /dev/raw/raw847  /dev/sdav12
raw /dev/raw/raw848  /dev/sdaw12
raw /dev/raw/raw849  /dev/sdax12
raw /dev/raw/raw850  /dev/sday12
raw /dev/raw/raw851  /dev/sdaz12
raw /dev/raw/raw852  /dev/sdba12
raw /dev/raw/raw853  /dev/sdbb12
raw /dev/raw/raw854  /dev/sdbc12
raw /dev/raw/raw855  /dev/sdbd12
raw /dev/raw/raw856  /dev/sdbe12
raw /dev/raw/raw857  /dev/sdbf12
raw /dev/raw/raw858  /dev/sdbg12
raw /dev/raw/raw859  /dev/sdbh12
raw /dev/raw/raw860  /dev/sdbi12
raw /dev/raw/raw861  /dev/sdbj12
raw /dev/raw/raw862  /dev/sdbk12
raw /dev/raw/raw863  /dev/sdbl12
raw /dev/raw/raw864  /dev/sdbm12
raw /dev/raw/raw865  /dev/sdbn12
raw /dev/raw/raw866  /dev/sdbo12
raw /dev/raw/raw867  /dev/sdbp12
raw /dev/raw/raw868  /dev/sdbq12
raw /dev/raw/raw869  /dev/sdbr12
raw /dev/raw/raw870  /dev/sdbs12
raw /dev/raw/raw871  /dev/sdbt12
raw /dev/raw/raw872  /dev/sdbu12
raw /dev/raw/raw873  /dev/sdbv12
raw /dev/raw/raw874  /dev/sdbw12
raw /dev/raw/raw875  /dev/sdbx12
raw /dev/raw/raw876  /dev/sdby12
raw /dev/raw/raw877  /dev/sdbz12
raw /dev/raw/raw878  /dev/sdca12
raw /dev/raw/raw879  /dev/sdcb12
raw /dev/raw/raw880  /dev/sdcc12

raw /dev/raw/raw881  /dev/sdb13
raw /dev/raw/raw882  /dev/sdc13
raw /dev/raw/raw883  /dev/sdd13
raw /dev/raw/raw884  /dev/sde13
raw /dev/raw/raw885  /dev/sdf13
raw /dev/raw/raw886  /dev/sdg13
raw /dev/raw/raw887  /dev/sdh13
raw /dev/raw/raw888  /dev/sdi13
raw /dev/raw/raw889  /dev/sdj13
raw /dev/raw/raw890  /dev/sdk13
raw /dev/raw/raw891  /dev/sdl13
raw /dev/raw/raw892  /dev/sdm13
raw /dev/raw/raw893  /dev/sdn13
raw /dev/raw/raw894  /dev/sdo13
raw /dev/raw/raw895  /dev/sdp13
raw /dev/raw/raw896  /dev/sdq13
raw /dev/raw/raw897  /dev/sdr13
raw /dev/raw/raw898  /dev/sds13
raw /dev/raw/raw899  /dev/sdt13
raw /dev/raw/raw900  /dev/sdu13
raw /dev/raw/raw901  /dev/sdv13
raw /dev/raw/raw902  /dev/sdw13
raw /dev/raw/raw903  /dev/sdx13
raw /dev/raw/raw904  /dev/sdy13
raw /dev/raw/raw905  /dev/sdz13
raw /dev/raw/raw906  /dev/sdaa13
raw /dev/raw/raw907  /dev/sdab13
raw /dev/raw/raw908  /dev/sdac13
raw /dev/raw/raw909  /dev/sdad13
raw /dev/raw/raw910  /dev/sdae13
raw /dev/raw/raw911  /dev/sdaf13
raw /dev/raw/raw912  /dev/sdag13
raw /dev/raw/raw913  /dev/sdah13
raw /dev/raw/raw914  /dev/sdai13
raw /dev/raw/raw915  /dev/sdaj13
raw /dev/raw/raw916  /dev/sdak13
raw /dev/raw/raw917  /dev/sdal13
raw /dev/raw/raw918  /dev/sdam13
raw /dev/raw/raw919  /dev/sdan13
```

```
raw /dev/raw/raw920     /dev/sdao13
raw /dev/raw/raw921     /dev/sdap13
raw /dev/raw/raw922     /dev/sdaq13
raw /dev/raw/raw923     /dev/sdar13
raw /dev/raw/raw924     /dev/sdas13
raw /dev/raw/raw925     /dev/sdat13
raw /dev/raw/raw926     /dev/sdau13
raw /dev/raw/raw927     /dev/sdav13
raw /dev/raw/raw928     /dev/sdaw13
raw /dev/raw/raw929     /dev/sdax13
raw /dev/raw/raw930     /dev/sday13
raw /dev/raw/raw931     /dev/sdaz13
raw /dev/raw/raw932     /dev/sdba13
raw /dev/raw/raw933     /dev/sdbb13
raw /dev/raw/raw934     /dev/sdbc13
raw /dev/raw/raw935     /dev/sdbd13
raw /dev/raw/raw936     /dev/sdbe13
raw /dev/raw/raw937     /dev/sdbf13
raw /dev/raw/raw938     /dev/sdbg13
raw /dev/raw/raw939     /dev/sdbh13
raw /dev/raw/raw940     /dev/sdbi13
raw /dev/raw/raw941     /dev/sdbj13
raw /dev/raw/raw942     /dev/sdbk13
raw /dev/raw/raw943     /dev/sdbl13
raw /dev/raw/raw944     /dev/sdbm13
raw /dev/raw/raw945     /dev/sdbn13
raw /dev/raw/raw946     /dev/sdbo13
raw /dev/raw/raw947     /dev/sdbp13
raw /dev/raw/raw948     /dev/sdbq13
raw /dev/raw/raw949     /dev/sdbr13
raw /dev/raw/raw950     /dev/sdbs13
raw /dev/raw/raw951     /dev/sdbt13
raw /dev/raw/raw952     /dev/sdbu13
raw /dev/raw/raw953     /dev/sdbv13
raw /dev/raw/raw954     /dev/sdbw13
raw /dev/raw/raw955     /dev/sdbx13
raw /dev/raw/raw956     /dev/sdby13
raw /dev/raw/raw957     /dev/sdbz13
raw /dev/raw/raw958     /dev/sdca13
raw /dev/raw/raw959     /dev/sdcb13
raw /dev/raw/raw960     /dev/sdcc13

raw /dev/raw/raw1300    /dev/sdcd

sleep 10

chmod 777 /dev/raw/raw*
```

version.txt

```
Linux itcopus82 2.6.16.21-0.8-smp #1 SMP Mon Jul 3 18:25:39 UTC 2006 x86_64 x86_64
x86_64 GNU/Linux
```

seteth.sh

```
ethtool -C eth9  tx-usecs 160 tx-frames 0 rx-usecs 300 rx-frames 0
ethtool -C eth11 tx-usecs 160 tx-frames 0 rx-usecs 300 rx-frames 0
```

11 Appendix C: Database Setup Code

11.1. Database Creation Scripts

DDL/ALTTBSP_PF_0.ddl

```
connect to TPCC;
alter tablespace is_customer_001 prefetchsize 0;
alter tablespace is_customer_002 prefetchsize 0;
alter tablespace is_customer_003 prefetchsize 0;
alter tablespace is_customer_004 prefetchsize 0;
alter tablespace is_customer_005 prefetchsize 0;
alter tablespace is_customer_006 prefetchsize 0;
alter tablespace is_customer_007 prefetchsize 0;
alter tablespace is_customer_008 prefetchsize 0;
alter tablespace is_customer_009 prefetchsize 0;
alter tablespace is_customer_010 prefetchsize 0;
alter tablespace is_customer_011 prefetchsize 0;
alter tablespace is_customer_012 prefetchsize 0;
alter tablespace is_customer_013 prefetchsize 0;
alter tablespace is_customer_014 prefetchsize 0;
alter tablespace is_customer_015 prefetchsize 0;
alter tablespace is_customer_016 prefetchsize 0;
alter tablespace is_customer_017 prefetchsize 0;
alter tablespace is_customer_018 prefetchsize 0;
alter tablespace is_customer_019 prefetchsize 0;
alter tablespace is_customer_020 prefetchsize 0;
alter tablespace is_customer_021 prefetchsize 0;
alter tablespace is_customer_022 prefetchsize 0;
alter tablespace is_customer_023 prefetchsize 0;
alter tablespace is_customer_024 prefetchsize 0;
alter tablespace is_customer_025 prefetchsize 0;
alter tablespace is_customer_026 prefetchsize 0;
alter tablespace is_customer_027 prefetchsize 0;
alter tablespace is_customer_028 prefetchsize 0;
alter tablespace is_customer_029 prefetchsize 0;
alter tablespace is_customer_030 prefetchsize 0;
alter tablespace is_customer_031 prefetchsize 0;
alter tablespace is_customer_032 prefetchsize 0;
alter tablespace is_customer_033 prefetchsize 0;
alter tablespace is_customer_034 prefetchsize 0;
alter tablespace is_customer_035 prefetchsize 0;
alter tablespace is_customer_036 prefetchsize 0;
alter tablespace is_customer_037 prefetchsize 0;
alter tablespace is_customer_038 prefetchsize 0;
alter tablespace is_customer_039 prefetchsize 0;
alter tablespace is_customer_040 prefetchsize 0;
alter tablespace is_customer_041 prefetchsize 0;
alter tablespace is_customer_042 prefetchsize 0;
alter tablespace is_customer_043 prefetchsize 0;
alter tablespace is_customer_044 prefetchsize 0;
alter tablespace is_customer_045 prefetchsize 0;
alter tablespace is_customer_046 prefetchsize 0;
alter tablespace is_customer_047 prefetchsize 0;
alter tablespace is_customer_048 prefetchsize 0;
alter tablespace is_customer_049 prefetchsize 0;
alter tablespace is_customer_050 prefetchsize 0;
alter tablespace is_customer_051 prefetchsize 0;
alter tablespace is_customer_052 prefetchsize 0;
alter tablespace is_customer_053 prefetchsize 0;
alter tablespace is_customer_054 prefetchsize 0;
alter tablespace is_customer_055 prefetchsize 0;
alter tablespace is_customer_056 prefetchsize 0;
alter tablespace is_customer_057 prefetchsize 0;
alter tablespace is_customer_058 prefetchsize 0;
alter tablespace is_customer_059 prefetchsize 0;
alter tablespace is_customer_060 prefetchsize 0;
alter tablespace is_customer_061 prefetchsize 0;
alter tablespace is_customer_062 prefetchsize 0;
alter tablespace is_customer_063 prefetchsize 0;
alter tablespace is_customer_064 prefetchsize 0;
alter tablespace is_customer_065 prefetchsize 0;
alter tablespace is_customer_066 prefetchsize 0;
alter tablespace is_customer_067 prefetchsize 0;
alter tablespace is_customer_068 prefetchsize 0;
alter tablespace is_customer_069 prefetchsize 0;
alter tablespace is_customer_070 prefetchsize 0;
alter tablespace is_customer_071 prefetchsize 0;
alter tablespace is_customer_072 prefetchsize 0;
alter tablespace is_customer_073 prefetchsize 0;
alter tablespace is_customer_074 prefetchsize 0;
alter tablespace is_customer_075 prefetchsize 0;
alter tablespace is_customer_076 prefetchsize 0;
alter tablespace is_customer_077 prefetchsize 0;
alter tablespace is_customer_078 prefetchsize 0;
alter tablespace is_customer_079 prefetchsize 0;
alter tablespace is_customer_080 prefetchsize 0;
alter tablespace is_order_001 prefetchsize 0;
alter tablespace is_order_002 prefetchsize 0;
alter tablespace is_order_003 prefetchsize 0;
alter tablespace is_order_004 prefetchsize 0;
alter tablespace is_order_005 prefetchsize 0;
alter tablespace is_order_006 prefetchsize 0;
alter tablespace is_order_007 prefetchsize 0;
alter tablespace is_order_008 prefetchsize 0;
alter tablespace is_order_009 prefetchsize 0;
alter tablespace is_order_010 prefetchsize 0;
alter tablespace is_order_011 prefetchsize 0;
alter tablespace is_order_012 prefetchsize 0;
alter tablespace is_order_013 prefetchsize 0;
alter tablespace is_order_014 prefetchsize 0;
alter tablespace is_order_015 prefetchsize 0;
alter tablespace is_order_016 prefetchsize 0;
alter tablespace is_order_017 prefetchsize 0;
alter tablespace is_order_018 prefetchsize 0;
alter tablespace is_order_019 prefetchsize 0;
alter tablespace is_order_020 prefetchsize 0;
alter tablespace is_order_021 prefetchsize 0;
alter tablespace is_order_022 prefetchsize 0;
alter tablespace is_order_023 prefetchsize 0;
alter tablespace is_order_024 prefetchsize 0;
alter tablespace is_order_025 prefetchsize 0;
alter tablespace is_order_026 prefetchsize 0;
alter tablespace is_order_027 prefetchsize 0;
alter tablespace is_order_028 prefetchsize 0;
alter tablespace is_order_029 prefetchsize 0;
alter tablespace is_order_030 prefetchsize 0;
alter tablespace is_order_031 prefetchsize 0;
alter tablespace is_order_032 prefetchsize 0;
alter tablespace is_order_033 prefetchsize 0;
alter tablespace is_order_034 prefetchsize 0;
alter tablespace is_order_035 prefetchsize 0;
alter tablespace is_order_036 prefetchsize 0;
alter tablespace is_order_037 prefetchsize 0;
alter tablespace is_order_038 prefetchsize 0;
alter tablespace is_order_039 prefetchsize 0;
alter tablespace is_order_040 prefetchsize 0;
alter tablespace is_order_041 prefetchsize 0;
alter tablespace is_order_042 prefetchsize 0;
alter tablespace is_order_043 prefetchsize 0;
alter tablespace is_order_044 prefetchsize 0;
alter tablespace is_order_045 prefetchsize 0;
alter tablespace is_order_046 prefetchsize 0;
alter tablespace is_order_047 prefetchsize 0;
alter tablespace is_order_048 prefetchsize 0;
alter tablespace is_order_049 prefetchsize 0;
alter tablespace is_order_050 prefetchsize 0;
alter tablespace is_order_051 prefetchsize 0;
alter tablespace is_order_052 prefetchsize 0;
alter tablespace is_order_053 prefetchsize 0;
alter tablespace is_order_054 prefetchsize 0;
alter tablespace is_order_055 prefetchsize 0;
alter tablespace is_order_056 prefetchsize 0;
alter tablespace is_order_057 prefetchsize 0;
alter tablespace is_order_058 prefetchsize 0;
alter tablespace is_order_059 prefetchsize 0;
alter tablespace is_order_060 prefetchsize 0;
alter tablespace is_order_061 prefetchsize 0;
alter tablespace is_order_062 prefetchsize 0;
alter tablespace is_order_063 prefetchsize 0;
alter tablespace is_order_064 prefetchsize 0;
alter tablespace is_order_065 prefetchsize 0;
alter tablespace is_order_066 prefetchsize 0;
alter tablespace is_order_067 prefetchsize 0;
alter tablespace is_order_068 prefetchsize 0;
alter tablespace is_order_069 prefetchsize 0;
alter tablespace is_order_070 prefetchsize 0;
alter tablespace is_order_071 prefetchsize 0;
alter tablespace is_order_072 prefetchsize 0;
alter tablespace is_order_073 prefetchsize 0;
alter tablespace is_order_074 prefetchsize 0;
alter tablespace is_order_075 prefetchsize 0;
alter tablespace is_order_076 prefetchsize 0;
alter tablespace is_order_077 prefetchsize 0;
alter tablespace is_order_078 prefetchsize 0;
alter tablespace is_order_079 prefetchsize 0;
alter tablespace is_order_080 prefetchsize 0;
alter tablespace ts_customer_001 prefetchsize 0;
alter tablespace ts_customer_002 prefetchsize 0;
alter tablespace ts_customer_003 prefetchsize 0;
alter tablespace ts_customer_004 prefetchsize 0;
alter tablespace ts_customer_005 prefetchsize 0;
alter tablespace ts_customer_006 prefetchsize 0;
alter tablespace ts_customer_007 prefetchsize 0;
alter tablespace ts_customer_008 prefetchsize 0;
alter tablespace ts_customer_009 prefetchsize 0;
alter tablespace ts_customer_010 prefetchsize 0;
alter tablespace ts_customer_011 prefetchsize 0;
alter tablespace ts_customer_012 prefetchsize 0;
alter tablespace ts_customer_013 prefetchsize 0;
alter tablespace ts_customer_014 prefetchsize 0;
alter tablespace ts_customer_015 prefetchsize 0;
alter tablespace ts_customer_016 prefetchsize 0;
alter tablespace ts_customer_017 prefetchsize 0;
alter tablespace ts_customer_018 prefetchsize 0;
alter tablespace ts_customer_019 prefetchsize 0;
alter tablespace ts_customer_020 prefetchsize 0;
alter tablespace ts_customer_021 prefetchsize 0;
alter tablespace ts_customer_022 prefetchsize 0;
alter tablespace ts_customer_023 prefetchsize 0;
alter tablespace ts_customer_024 prefetchsize 0;
alter tablespace ts_customer_025 prefetchsize 0;
alter tablespace ts_customer_026 prefetchsize 0;
alter tablespace ts_customer_027 prefetchsize 0;
alter tablespace ts_customer_028 prefetchsize 0;
alter tablespace ts_customer_029 prefetchsize 0;
alter tablespace ts_customer_030 prefetchsize 0;
alter tablespace ts_customer_031 prefetchsize 0;
alter tablespace ts_customer_032 prefetchsize 0;
alter tablespace ts_customer_033 prefetchsize 0;
```

```
alter tablespace ts_customer_034 prefetchsize 0;
alter tablespace ts_customer_035 prefetchsize 0;
alter tablespace ts_customer_036 prefetchsize 0;
alter tablespace ts_customer_037 prefetchsize 0;
alter tablespace ts_customer_038 prefetchsize 0;
alter tablespace ts_customer_039 prefetchsize 0;
alter tablespace ts_customer_040 prefetchsize 0;
alter tablespace ts_customer_041 prefetchsize 0;
alter tablespace ts_customer_042 prefetchsize 0;
alter tablespace ts_customer_043 prefetchsize 0;
alter tablespace ts_customer_044 prefetchsize 0;
alter tablespace ts_customer_045 prefetchsize 0;
alter tablespace ts_customer_046 prefetchsize 0;
alter tablespace ts_customer_047 prefetchsize 0;
alter tablespace ts_customer_048 prefetchsize 0;
alter tablespace ts_customer_049 prefetchsize 0;
alter tablespace ts_customer_050 prefetchsize 0;
alter tablespace ts_customer_051 prefetchsize 0;
alter tablespace ts_customer_052 prefetchsize 0;
alter tablespace ts_customer_053 prefetchsize 0;
alter tablespace ts_customer_054 prefetchsize 0;
alter tablespace ts_customer_055 prefetchsize 0;
alter tablespace ts_customer_056 prefetchsize 0;
alter tablespace ts_customer_057 prefetchsize 0;
alter tablespace ts_customer_058 prefetchsize 0;
alter tablespace ts_customer_059 prefetchsize 0;
alter tablespace ts_customer_060 prefetchsize 0;
alter tablespace ts_customer_061 prefetchsize 0;
alter tablespace ts_customer_062 prefetchsize 0;
alter tablespace ts_customer_063 prefetchsize 0;
alter tablespace ts_customer_064 prefetchsize 0;
alter tablespace ts_customer_065 prefetchsize 0;
alter tablespace ts_customer_066 prefetchsize 0;
alter tablespace ts_customer_067 prefetchsize 0;
alter tablespace ts_customer_068 prefetchsize 0;
alter tablespace ts_customer_069 prefetchsize 0;
alter tablespace ts_customer_070 prefetchsize 0;
alter tablespace ts_customer_071 prefetchsize 0;
alter tablespace ts_customer_072 prefetchsize 0;
alter tablespace ts_customer_073 prefetchsize 0;
alter tablespace ts_customer_074 prefetchsize 0;
alter tablespace ts_customer_075 prefetchsize 0;
alter tablespace ts_customer_076 prefetchsize 0;
alter tablespace ts_customer_077 prefetchsize 0;
alter tablespace ts_customer_078 prefetchsize 0;
alter tablespace ts_customer_079 prefetchsize 0;
alter tablespace ts_customer_080 prefetchsize 0;
alter tablespace ts_dis_001 prefetchsize 0;
alter tablespace ts_dis_002 prefetchsize 0;
alter tablespace ts_dis_003 prefetchsize 0;
alter tablespace ts_dis_004 prefetchsize 0;
alter tablespace ts_dis_005 prefetchsize 0;
alter tablespace ts_dis_006 prefetchsize 0;
alter tablespace ts_dis_007 prefetchsize 0;
alter tablespace ts_dis_008 prefetchsize 0;
alter tablespace ts_dis_009 prefetchsize 0;
alter tablespace ts_dis_010 prefetchsize 0;
alter tablespace ts_dis_011 prefetchsize 0;
alter tablespace ts_dis_012 prefetchsize 0;
alter tablespace ts_dis_013 prefetchsize 0;
alter tablespace ts_dis_014 prefetchsize 0;
alter tablespace ts_dis_015 prefetchsize 0;
alter tablespace ts_dis_016 prefetchsize 0;
alter tablespace ts_dis_017 prefetchsize 0;
alter tablespace ts_dis_018 prefetchsize 0;
alter tablespace ts_dis_019 prefetchsize 0;
alter tablespace ts_dis_020 prefetchsize 0;
alter tablespace ts_dis_021 prefetchsize 0;
alter tablespace ts_dis_022 prefetchsize 0;
alter tablespace ts_dis_023 prefetchsize 0;
alter tablespace ts_dis_024 prefetchsize 0;
alter tablespace ts_dis_025 prefetchsize 0;
alter tablespace ts_dis_026 prefetchsize 0;
alter tablespace ts_dis_027 prefetchsize 0;
alter tablespace ts_dis_028 prefetchsize 0;
alter tablespace ts_dis_029 prefetchsize 0;
alter tablespace ts_dis_030 prefetchsize 0;
alter tablespace ts_dis_031 prefetchsize 0;
alter tablespace ts_dis_032 prefetchsize 0;
alter tablespace ts_dis_033 prefetchsize 0;
alter tablespace ts_dis_034 prefetchsize 0;
alter tablespace ts_dis_035 prefetchsize 0;
alter tablespace ts_dis_036 prefetchsize 0;
alter tablespace ts_dis_037 prefetchsize 0;
alter tablespace ts_dis_038 prefetchsize 0;
alter tablespace ts_dis_039 prefetchsize 0;
alter tablespace ts_dis_040 prefetchsize 0;
alter tablespace ts_dis_041 prefetchsize 0;
alter tablespace ts_dis_042 prefetchsize 0;
alter tablespace ts_dis_043 prefetchsize 0;
alter tablespace ts_dis_044 prefetchsize 0;
alter tablespace ts_dis_045 prefetchsize 0;
alter tablespace ts_dis_046 prefetchsize 0;
alter tablespace ts_dis_047 prefetchsize 0;
alter tablespace ts_dis_048 prefetchsize 0;
alter tablespace ts_dis_049 prefetchsize 0;
alter tablespace ts_dis_050 prefetchsize 0;
alter tablespace ts_dis_051 prefetchsize 0;
alter tablespace ts_dis_052 prefetchsize 0;
alter tablespace ts_dis_053 prefetchsize 0;
alter tablespace ts_dis_054 prefetchsize 0;
alter tablespace ts_dis_055 prefetchsize 0;
alter tablespace ts_dis_056 prefetchsize 0;
alter tablespace ts_dis_057 prefetchsize 0;
alter tablespace ts_dis_058 prefetchsize 0;
alter tablespace ts_dis_059 prefetchsize 0;
alter tablespace ts_dis_060 prefetchsize 0;
alter tablespace ts_dis_061 prefetchsize 0;
alter tablespace ts_dis_062 prefetchsize 0;
alter tablespace ts_dis_063 prefetchsize 0;
alter tablespace ts_dis_064 prefetchsize 0;
alter tablespace ts_dis_065 prefetchsize 0;
alter tablespace ts_dis_066 prefetchsize 0;
alter tablespace ts_dis_067 prefetchsize 0;
alter tablespace ts_dis_068 prefetchsize 0;
alter tablespace ts_dis_069 prefetchsize 0;
alter tablespace ts_dis_070 prefetchsize 0;
alter tablespace ts_dis_071 prefetchsize 0;
alter tablespace ts_dis_072 prefetchsize 0;
alter tablespace ts_dis_073 prefetchsize 0;
alter tablespace ts_dis_074 prefetchsize 0;
alter tablespace ts_dis_075 prefetchsize 0;
alter tablespace ts_dis_076 prefetchsize 0;
alter tablespace ts_dis_077 prefetchsize 0;
alter tablespace ts_dis_078 prefetchsize 0;
alter tablespace ts_dis_079 prefetchsize 0;
alter tablespace ts_dis_080 prefetchsize 0;
alter tablespace ts_history_001 prefetchsize 0;
alter tablespace ts_history_002 prefetchsize 0;
alter tablespace ts_history_003 prefetchsize 0;
alter tablespace ts_history_004 prefetchsize 0;
alter tablespace ts_history_005 prefetchsize 0;
alter tablespace ts_history_006 prefetchsize 0;
alter tablespace ts_history_007 prefetchsize 0;
alter tablespace ts_history_008 prefetchsize 0;
alter tablespace ts_history_009 prefetchsize 0;
alter tablespace ts_history_010 prefetchsize 0;
alter tablespace ts_history_011 prefetchsize 0;
alter tablespace ts_history_012 prefetchsize 0;
alter tablespace ts_history_013 prefetchsize 0;
alter tablespace ts_history_014 prefetchsize 0;
alter tablespace ts_history_015 prefetchsize 0;
alter tablespace ts_history_016 prefetchsize 0;
alter tablespace ts_history_017 prefetchsize 0;
alter tablespace ts_history_018 prefetchsize 0;
alter tablespace ts_history_019 prefetchsize 0;
alter tablespace ts_history_020 prefetchsize 0;
alter tablespace ts_history_021 prefetchsize 0;
alter tablespace ts_history_022 prefetchsize 0;
alter tablespace ts_history_023 prefetchsize 0;
alter tablespace ts_history_024 prefetchsize 0;
alter tablespace ts_history_025 prefetchsize 0;
alter tablespace ts_history_026 prefetchsize 0;
alter tablespace ts_history_027 prefetchsize 0;
alter tablespace ts_history_028 prefetchsize 0;
alter tablespace ts_history_029 prefetchsize 0;
alter tablespace ts_history_030 prefetchsize 0;
alter tablespace ts_history_031 prefetchsize 0;
alter tablespace ts_history_032 prefetchsize 0;
alter tablespace ts_history_033 prefetchsize 0;
alter tablespace ts_history_034 prefetchsize 0;
alter tablespace ts_history_035 prefetchsize 0;
alter tablespace ts_history_036 prefetchsize 0;
alter tablespace ts_history_037 prefetchsize 0;
alter tablespace ts_history_038 prefetchsize 0;
alter tablespace ts_history_039 prefetchsize 0;
alter tablespace ts_history_040 prefetchsize 0;
alter tablespace ts_history_041 prefetchsize 0;
alter tablespace ts_history_042 prefetchsize 0;
alter tablespace ts_history_043 prefetchsize 0;
alter tablespace ts_history_044 prefetchsize 0;
alter tablespace ts_history_045 prefetchsize 0;
alter tablespace ts_history_046 prefetchsize 0;
alter tablespace ts_history_047 prefetchsize 0;
alter tablespace ts_history_048 prefetchsize 0;
alter tablespace ts_history_049 prefetchsize 0;
alter tablespace ts_history_050 prefetchsize 0;
alter tablespace ts_history_051 prefetchsize 0;
alter tablespace ts_history_052 prefetchsize 0;
alter tablespace ts_history_053 prefetchsize 0;
alter tablespace ts_history_054 prefetchsize 0;
alter tablespace ts_history_055 prefetchsize 0;
alter tablespace ts_history_056 prefetchsize 0;
alter tablespace ts_history_057 prefetchsize 0;
alter tablespace ts_history_058 prefetchsize 0;
alter tablespace ts_history_059 prefetchsize 0;
alter tablespace ts_history_060 prefetchsize 0;
alter tablespace ts_history_061 prefetchsize 0;
alter tablespace ts_history_062 prefetchsize 0;
alter tablespace ts_history_063 prefetchsize 0;
alter tablespace ts_history_064 prefetchsize 0;
alter tablespace ts_history_065 prefetchsize 0;
alter tablespace ts_history_066 prefetchsize 0;
alter tablespace ts_history_067 prefetchsize 0;
alter tablespace ts_history_068 prefetchsize 0;
alter tablespace ts_history_069 prefetchsize 0;
alter tablespace ts_history_070 prefetchsize 0;
alter tablespace ts_history_071 prefetchsize 0;
alter tablespace ts_history_072 prefetchsize 0;
alter tablespace ts_history_073 prefetchsize 0;
alter tablespace ts_history_074 prefetchsize 0;
alter tablespace ts_history_075 prefetchsize 0;
alter tablespace ts_history_076 prefetchsize 0;
alter tablespace ts_history_077 prefetchsize 0;
alter tablespace ts_history_078 prefetchsize 0;
alter tablespace ts_history_079 prefetchsize 0;
alter tablespace ts_history_080 prefetchsize 0;
alter tablespace ts_item prefetchsize 0;
alter tablespace ts_newordA_001 prefetchsize 0;
alter tablespace ts_newordA_002 prefetchsize 0;
```

```
alter tablespace ts_newordA_003 prefetchsize 0;
alter tablespace ts_newordA_004 prefetchsize 0;
alter tablespace ts_newordA_005 prefetchsize 0;
alter tablespace ts_newordA_006 prefetchsize 0;
alter tablespace ts_newordA_007 prefetchsize 0;
alter tablespace ts_newordA_008 prefetchsize 0;
alter tablespace ts_newordA_009 prefetchsize 0;
alter tablespace ts_newordA_010 prefetchsize 0;
alter tablespace ts_newordA_011 prefetchsize 0;
alter tablespace ts_newordA_012 prefetchsize 0;
alter tablespace ts_newordA_013 prefetchsize 0;
alter tablespace ts_newordA_014 prefetchsize 0;
alter tablespace ts_newordA_015 prefetchsize 0;
alter tablespace ts_newordA_016 prefetchsize 0;
alter tablespace ts_newordA_017 prefetchsize 0;
alter tablespace ts_newordA_018 prefetchsize 0;
alter tablespace ts_newordA_019 prefetchsize 0;
alter tablespace ts_newordA_020 prefetchsize 0;
alter tablespace ts_newordA_021 prefetchsize 0;
alter tablespace ts_newordA_022 prefetchsize 0;
alter tablespace ts_newordA_023 prefetchsize 0;
alter tablespace ts_newordA_024 prefetchsize 0;
alter tablespace ts_newordA_025 prefetchsize 0;
alter tablespace ts_newordA_026 prefetchsize 0;
alter tablespace ts_newordA_027 prefetchsize 0;
alter tablespace ts_newordA_028 prefetchsize 0;
alter tablespace ts_newordA_029 prefetchsize 0;
alter tablespace ts_newordA_030 prefetchsize 0;
alter tablespace ts_newordA_031 prefetchsize 0;
alter tablespace ts_newordA_032 prefetchsize 0;
alter tablespace ts_newordA_033 prefetchsize 0;
alter tablespace ts_newordA_034 prefetchsize 0;
alter tablespace ts_newordA_035 prefetchsize 0;
alter tablespace ts_newordA_036 prefetchsize 0;
alter tablespace ts_newordA_037 prefetchsize 0;
alter tablespace ts_newordA_038 prefetchsize 0;
alter tablespace ts_newordA_039 prefetchsize 0;
alter tablespace ts_newordA_040 prefetchsize 0;
alter tablespace ts_newordA_041 prefetchsize 0;
alter tablespace ts_newordA_042 prefetchsize 0;
alter tablespace ts_newordA_043 prefetchsize 0;
alter tablespace ts_newordA_044 prefetchsize 0;
alter tablespace ts_newordA_045 prefetchsize 0;
alter tablespace ts_newordA_046 prefetchsize 0;
alter tablespace ts_newordA_047 prefetchsize 0;
alter tablespace ts_newordA_048 prefetchsize 0;
alter tablespace ts_newordA_049 prefetchsize 0;
alter tablespace ts_newordA_050 prefetchsize 0;
alter tablespace ts_newordA_051 prefetchsize 0;
alter tablespace ts_newordA_052 prefetchsize 0;
alter tablespace ts_newordA_053 prefetchsize 0;
alter tablespace ts_newordA_054 prefetchsize 0;
alter tablespace ts_newordA_055 prefetchsize 0;
alter tablespace ts_newordA_056 prefetchsize 0;
alter tablespace ts_newordA_057 prefetchsize 0;
alter tablespace ts_newordA_058 prefetchsize 0;
alter tablespace ts_newordA_059 prefetchsize 0;
alter tablespace ts_newordA_060 prefetchsize 0;
alter tablespace ts_newordA_061 prefetchsize 0;
alter tablespace ts_newordA_062 prefetchsize 0;
alter tablespace ts_newordA_063 prefetchsize 0;
alter tablespace ts_newordA_064 prefetchsize 0;
alter tablespace ts_newordA_065 prefetchsize 0;
alter tablespace ts_newordA_066 prefetchsize 0;
alter tablespace ts_newordA_067 prefetchsize 0;
alter tablespace ts_newordA_068 prefetchsize 0;
alter tablespace ts_newordA_069 prefetchsize 0;
alter tablespace ts_newordA_070 prefetchsize 0;
alter tablespace ts_newordA_071 prefetchsize 0;
alter tablespace ts_newordA_072 prefetchsize 0;
alter tablespace ts_newordA_073 prefetchsize 0;
alter tablespace ts_newordA_074 prefetchsize 0;
alter tablespace ts_newordA_075 prefetchsize 0;
alter tablespace ts_newordA_076 prefetchsize 0;
alter tablespace ts_newordA_077 prefetchsize 0;
alter tablespace ts_newordA_078 prefetchsize 0;
alter tablespace ts_newordA_079 prefetchsize 0;
alter tablespace ts_newordA_080 prefetchsize 0;
alter tablespace ts_newordB_001 prefetchsize 0;
alter tablespace ts_newordB_002 prefetchsize 0;
alter tablespace ts_newordB_003 prefetchsize 0;
alter tablespace ts_newordB_004 prefetchsize 0;
alter tablespace ts_newordB_005 prefetchsize 0;
alter tablespace ts_newordB_006 prefetchsize 0;
alter tablespace ts_newordB_007 prefetchsize 0;
alter tablespace ts_newordB_008 prefetchsize 0;
alter tablespace ts_newordB_009 prefetchsize 0;
alter tablespace ts_newordB_010 prefetchsize 0;
alter tablespace ts_newordB_011 prefetchsize 0;
alter tablespace ts_newordB_012 prefetchsize 0;
alter tablespace ts_newordB_013 prefetchsize 0;
alter tablespace ts_newordB_014 prefetchsize 0;
alter tablespace ts_newordB_015 prefetchsize 0;
alter tablespace ts_newordB_016 prefetchsize 0;
alter tablespace ts_newordB_017 prefetchsize 0;
alter tablespace ts_newordB_018 prefetchsize 0;
alter tablespace ts_newordB_019 prefetchsize 0;
alter tablespace ts_newordB_020 prefetchsize 0;
alter tablespace ts_newordB_021 prefetchsize 0;
alter tablespace ts_newordB_022 prefetchsize 0;
alter tablespace ts_newordB_023 prefetchsize 0;
alter tablespace ts_newordB_024 prefetchsize 0;
alter tablespace ts_newordB_025 prefetchsize 0;
alter tablespace ts_newordB_026 prefetchsize 0;
alter tablespace ts_newordB_027 prefetchsize 0;
alter tablespace ts_newordB_028 prefetchsize 0;
alter tablespace ts_newordB_029 prefetchsize 0;
alter tablespace ts_newordB_030 prefetchsize 0;
alter tablespace ts_newordB_031 prefetchsize 0;
alter tablespace ts_newordB_032 prefetchsize 0;
alter tablespace ts_newordB_033 prefetchsize 0;
alter tablespace ts_newordB_034 prefetchsize 0;
alter tablespace ts_newordB_035 prefetchsize 0;
alter tablespace ts_newordB_036 prefetchsize 0;
alter tablespace ts_newordB_037 prefetchsize 0;
alter tablespace ts_newordB_038 prefetchsize 0;
alter tablespace ts_newordB_039 prefetchsize 0;
alter tablespace ts_newordB_040 prefetchsize 0;
alter tablespace ts_newordB_041 prefetchsize 0;
alter tablespace ts_newordB_042 prefetchsize 0;
alter tablespace ts_newordB_043 prefetchsize 0;
alter tablespace ts_newordB_044 prefetchsize 0;
alter tablespace ts_newordB_045 prefetchsize 0;
alter tablespace ts_newordB_046 prefetchsize 0;
alter tablespace ts_newordB_047 prefetchsize 0;
alter tablespace ts_newordB_048 prefetchsize 0;
alter tablespace ts_newordB_049 prefetchsize 0;
alter tablespace ts_newordB_050 prefetchsize 0;
alter tablespace ts_newordB_051 prefetchsize 0;
alter tablespace ts_newordB_052 prefetchsize 0;
alter tablespace ts_newordB_053 prefetchsize 0;
alter tablespace ts_newordB_054 prefetchsize 0;
alter tablespace ts_newordB_055 prefetchsize 0;
alter tablespace ts_newordB_056 prefetchsize 0;
alter tablespace ts_newordB_057 prefetchsize 0;
alter tablespace ts_newordB_058 prefetchsize 0;
alter tablespace ts_newordB_059 prefetchsize 0;
alter tablespace ts_newordB_060 prefetchsize 0;
alter tablespace ts_newordB_061 prefetchsize 0;
alter tablespace ts_newordB_062 prefetchsize 0;
alter tablespace ts_newordB_063 prefetchsize 0;
alter tablespace ts_newordB_064 prefetchsize 0;
alter tablespace ts_newordB_065 prefetchsize 0;
alter tablespace ts_newordB_066 prefetchsize 0;
alter tablespace ts_newordB_067 prefetchsize 0;
alter tablespace ts_newordB_068 prefetchsize 0;
alter tablespace ts_newordB_069 prefetchsize 0;
alter tablespace ts_newordB_070 prefetchsize 0;
alter tablespace ts_newordB_071 prefetchsize 0;
alter tablespace ts_newordB_072 prefetchsize 0;
alter tablespace ts_newordB_073 prefetchsize 0;
alter tablespace ts_newordB_074 prefetchsize 0;
alter tablespace ts_newordB_075 prefetchsize 0;
alter tablespace ts_newordB_076 prefetchsize 0;
alter tablespace ts_newordB_077 prefetchsize 0;
alter tablespace ts_newordB_078 prefetchsize 0;
alter tablespace ts_newordB_079 prefetchsize 0;
alter tablespace ts_newordB_080 prefetchsize 0;
alter tablespace ts_order_001 prefetchsize 0;
alter tablespace ts_order_002 prefetchsize 0;
alter tablespace ts_order_003 prefetchsize 0;
alter tablespace ts_order_004 prefetchsize 0;
alter tablespace ts_order_005 prefetchsize 0;
alter tablespace ts_order_006 prefetchsize 0;
alter tablespace ts_order_007 prefetchsize 0;
alter tablespace ts_order_008 prefetchsize 0;
alter tablespace ts_order_009 prefetchsize 0;
alter tablespace ts_order_010 prefetchsize 0;
alter tablespace ts_order_011 prefetchsize 0;
alter tablespace ts_order_012 prefetchsize 0;
alter tablespace ts_order_013 prefetchsize 0;
alter tablespace ts_order_014 prefetchsize 0;
alter tablespace ts_order_015 prefetchsize 0;
alter tablespace ts_order_016 prefetchsize 0;
alter tablespace ts_order_017 prefetchsize 0;
alter tablespace ts_order_018 prefetchsize 0;
alter tablespace ts_order_019 prefetchsize 0;
alter tablespace ts_order_020 prefetchsize 0;
alter tablespace ts_order_021 prefetchsize 0;
alter tablespace ts_order_022 prefetchsize 0;
alter tablespace ts_order_023 prefetchsize 0;
alter tablespace ts_order_024 prefetchsize 0;
alter tablespace ts_order_025 prefetchsize 0;
alter tablespace ts_order_026 prefetchsize 0;
alter tablespace ts_order_027 prefetchsize 0;
alter tablespace ts_order_028 prefetchsize 0;
alter tablespace ts_order_029 prefetchsize 0;
alter tablespace ts_order_030 prefetchsize 0;
alter tablespace ts_order_031 prefetchsize 0;
alter tablespace ts_order_032 prefetchsize 0;
alter tablespace ts_order_033 prefetchsize 0;
alter tablespace ts_order_034 prefetchsize 0;
alter tablespace ts_order_035 prefetchsize 0;
alter tablespace ts_order_036 prefetchsize 0;
alter tablespace ts_order_037 prefetchsize 0;
alter tablespace ts_order_038 prefetchsize 0;
alter tablespace ts_order_039 prefetchsize 0;
alter tablespace ts_order_040 prefetchsize 0;
alter tablespace ts_order_041 prefetchsize 0;
alter tablespace ts_order_042 prefetchsize 0;
alter tablespace ts_order_043 prefetchsize 0;
alter tablespace ts_order_044 prefetchsize 0;
alter tablespace ts_order_045 prefetchsize 0;
alter tablespace ts_order_046 prefetchsize 0;
alter tablespace ts_order_047 prefetchsize 0;
alter tablespace ts_order_048 prefetchsize 0;
alter tablespace ts_order_049 prefetchsize 0;
alter tablespace ts_order_050 prefetchsize 0;
alter tablespace ts_order_051 prefetchsize 0;
alter tablespace ts_order_052 prefetchsize 0;
```

```
alter tablespace ts_order_053 prefetchsize 0;
alter tablespace ts_order_054 prefetchsize 0;
alter tablespace ts_order_055 prefetchsize 0;
alter tablespace ts_order_056 prefetchsize 0;
alter tablespace ts_order_057 prefetchsize 0;
alter tablespace ts_order_058 prefetchsize 0;
alter tablespace ts_order_059 prefetchsize 0;
alter tablespace ts_order_060 prefetchsize 0;
alter tablespace ts_order_061 prefetchsize 0;
alter tablespace ts_order_062 prefetchsize 0;
alter tablespace ts_order_063 prefetchsize 0;
alter tablespace ts_order_064 prefetchsize 0;
alter tablespace ts_order_065 prefetchsize 0;
alter tablespace ts_order_066 prefetchsize 0;
alter tablespace ts_order_067 prefetchsize 0;
alter tablespace ts_order_068 prefetchsize 0;
alter tablespace ts_order_069 prefetchsize 0;
alter tablespace ts_order_070 prefetchsize 0;
alter tablespace ts_order_071 prefetchsize 0;
alter tablespace ts_order_072 prefetchsize 0;
alter tablespace ts_order_073 prefetchsize 0;
alter tablespace ts_order_074 prefetchsize 0;
alter tablespace ts_order_075 prefetchsize 0;
alter tablespace ts_order_076 prefetchsize 0;
alter tablespace ts_order_077 prefetchsize 0;
alter tablespace ts_order_078 prefetchsize 0;
alter tablespace ts_order_079 prefetchsize 0;
alter tablespace ts_order_080 prefetchsize 0;
alter tablespace ts_orderline_001 prefetchsize 0;
alter tablespace ts_orderline_002 prefetchsize 0;
alter tablespace ts_orderline_003 prefetchsize 0;
alter tablespace ts_orderline_004 prefetchsize 0;
alter tablespace ts_orderline_005 prefetchsize 0;
alter tablespace ts_orderline_006 prefetchsize 0;
alter tablespace ts_orderline_007 prefetchsize 0;
alter tablespace ts_orderline_008 prefetchsize 0;
alter tablespace ts_orderline_009 prefetchsize 0;
alter tablespace ts_orderline_010 prefetchsize 0;
alter tablespace ts_orderline_011 prefetchsize 0;
alter tablespace ts_orderline_012 prefetchsize 0;
alter tablespace ts_orderline_013 prefetchsize 0;
alter tablespace ts_orderline_014 prefetchsize 0;
alter tablespace ts_orderline_015 prefetchsize 0;
alter tablespace ts_orderline_016 prefetchsize 0;
alter tablespace ts_orderline_017 prefetchsize 0;
alter tablespace ts_orderline_018 prefetchsize 0;
alter tablespace ts_orderline_019 prefetchsize 0;
alter tablespace ts_orderline_020 prefetchsize 0;
alter tablespace ts_orderline_021 prefetchsize 0;
alter tablespace ts_orderline_022 prefetchsize 0;
alter tablespace ts_orderline_023 prefetchsize 0;
alter tablespace ts_orderline_024 prefetchsize 0;
alter tablespace ts_orderline_025 prefetchsize 0;
alter tablespace ts_orderline_026 prefetchsize 0;
alter tablespace ts_orderline_027 prefetchsize 0;
alter tablespace ts_orderline_028 prefetchsize 0;
alter tablespace ts_orderline_029 prefetchsize 0;
alter tablespace ts_orderline_030 prefetchsize 0;
alter tablespace ts_orderline_031 prefetchsize 0;
alter tablespace ts_orderline_032 prefetchsize 0;
alter tablespace ts_orderline_033 prefetchsize 0;
alter tablespace ts_orderline_034 prefetchsize 0;
alter tablespace ts_orderline_035 prefetchsize 0;
alter tablespace ts_orderline_036 prefetchsize 0;
alter tablespace ts_orderline_037 prefetchsize 0;
alter tablespace ts_orderline_038 prefetchsize 0;
alter tablespace ts_orderline_039 prefetchsize 0;
alter tablespace ts_orderline_040 prefetchsize 0;
alter tablespace ts_orderline_041 prefetchsize 0;
alter tablespace ts_orderline_042 prefetchsize 0;
alter tablespace ts_orderline_043 prefetchsize 0;
alter tablespace ts_orderline_044 prefetchsize 0;
alter tablespace ts_orderline_045 prefetchsize 0;
alter tablespace ts_orderline_046 prefetchsize 0;
alter tablespace ts_orderline_047 prefetchsize 0;
alter tablespace ts_orderline_048 prefetchsize 0;
alter tablespace ts_orderline_049 prefetchsize 0;
alter tablespace ts_orderline_050 prefetchsize 0;
alter tablespace ts_orderline_051 prefetchsize 0;
alter tablespace ts_orderline_052 prefetchsize 0;
alter tablespace ts_orderline_053 prefetchsize 0;
alter tablespace ts_orderline_054 prefetchsize 0;
alter tablespace ts_orderline_055 prefetchsize 0;
alter tablespace ts_orderline_056 prefetchsize 0;
alter tablespace ts_orderline_057 prefetchsize 0;
alter tablespace ts_orderline_058 prefetchsize 0;
alter tablespace ts_orderline_059 prefetchsize 0;
alter tablespace ts_orderline_060 prefetchsize 0;
alter tablespace ts_orderline_061 prefetchsize 0;
alter tablespace ts_orderline_062 prefetchsize 0;
alter tablespace ts_orderline_063 prefetchsize 0;
alter tablespace ts_orderline_064 prefetchsize 0;
alter tablespace ts_orderline_065 prefetchsize 0;
alter tablespace ts_orderline_066 prefetchsize 0;
alter tablespace ts_orderline_067 prefetchsize 0;
alter tablespace ts_orderline_068 prefetchsize 0;
alter tablespace ts_orderline_069 prefetchsize 0;
alter tablespace ts_orderline_070 prefetchsize 0;
alter tablespace ts_orderline_071 prefetchsize 0;
alter tablespace ts_orderline_072 prefetchsize 0;
alter tablespace ts_orderline_073 prefetchsize 0;
alter tablespace ts_orderline_074 prefetchsize 0;
alter tablespace ts_orderline_075 prefetchsize 0;
alter tablespace ts_orderline_076 prefetchsize 0;
alter tablespace ts_orderline_077 prefetchsize 0;
alter tablespace ts_orderline_078 prefetchsize 0;
alter tablespace ts_orderline_079 prefetchsize 0;
alter tablespace ts_orderline_080 prefetchsize 0;
alter tablespace ts_stock_001 prefetchsize 0;
alter tablespace ts_stock_002 prefetchsize 0;
alter tablespace ts_stock_003 prefetchsize 0;
alter tablespace ts_stock_004 prefetchsize 0;
alter tablespace ts_stock_005 prefetchsize 0;
alter tablespace ts_stock_006 prefetchsize 0;
alter tablespace ts_stock_007 prefetchsize 0;
alter tablespace ts_stock_008 prefetchsize 0;
alter tablespace ts_stock_009 prefetchsize 0;
alter tablespace ts_stock_010 prefetchsize 0;
alter tablespace ts_stock_011 prefetchsize 0;
alter tablespace ts_stock_012 prefetchsize 0;
alter tablespace ts_stock_013 prefetchsize 0;
alter tablespace ts_stock_014 prefetchsize 0;
alter tablespace ts_stock_015 prefetchsize 0;
alter tablespace ts_stock_016 prefetchsize 0;
alter tablespace ts_stock_017 prefetchsize 0;
alter tablespace ts_stock_018 prefetchsize 0;
alter tablespace ts_stock_019 prefetchsize 0;
alter tablespace ts_stock_020 prefetchsize 0;
alter tablespace ts_stock_021 prefetchsize 0;
alter tablespace ts_stock_022 prefetchsize 0;
alter tablespace ts_stock_023 prefetchsize 0;
alter tablespace ts_stock_024 prefetchsize 0;
alter tablespace ts_stock_025 prefetchsize 0;
alter tablespace ts_stock_026 prefetchsize 0;
alter tablespace ts_stock_027 prefetchsize 0;
alter tablespace ts_stock_028 prefetchsize 0;
alter tablespace ts_stock_029 prefetchsize 0;
alter tablespace ts_stock_030 prefetchsize 0;
alter tablespace ts_stock_031 prefetchsize 0;
alter tablespace ts_stock_032 prefetchsize 0;
alter tablespace ts_stock_033 prefetchsize 0;
alter tablespace ts_stock_034 prefetchsize 0;
alter tablespace ts_stock_035 prefetchsize 0;
alter tablespace ts_stock_036 prefetchsize 0;
alter tablespace ts_stock_037 prefetchsize 0;
alter tablespace ts_stock_038 prefetchsize 0;
alter tablespace ts_stock_039 prefetchsize 0;
alter tablespace ts_stock_040 prefetchsize 0;
alter tablespace ts_stock_041 prefetchsize 0;
alter tablespace ts_stock_042 prefetchsize 0;
alter tablespace ts_stock_043 prefetchsize 0;
alter tablespace ts_stock_044 prefetchsize 0;
alter tablespace ts_stock_045 prefetchsize 0;
alter tablespace ts_stock_046 prefetchsize 0;
alter tablespace ts_stock_047 prefetchsize 0;
alter tablespace ts_stock_048 prefetchsize 0;
alter tablespace ts_stock_049 prefetchsize 0;
alter tablespace ts_stock_050 prefetchsize 0;
alter tablespace ts_stock_051 prefetchsize 0;
alter tablespace ts_stock_052 prefetchsize 0;
alter tablespace ts_stock_053 prefetchsize 0;
alter tablespace ts_stock_054 prefetchsize 0;
alter tablespace ts_stock_055 prefetchsize 0;
alter tablespace ts_stock_056 prefetchsize 0;
alter tablespace ts_stock_057 prefetchsize 0;
alter tablespace ts_stock_058 prefetchsize 0;
alter tablespace ts_stock_059 prefetchsize 0;
alter tablespace ts_stock_060 prefetchsize 0;
alter tablespace ts_stock_061 prefetchsize 0;
alter tablespace ts_stock_062 prefetchsize 0;
alter tablespace ts_stock_063 prefetchsize 0;
alter tablespace ts_stock_064 prefetchsize 0;
alter tablespace ts_stock_065 prefetchsize 0;
alter tablespace ts_stock_066 prefetchsize 0;
alter tablespace ts_stock_067 prefetchsize 0;
alter tablespace ts_stock_068 prefetchsize 0;
alter tablespace ts_stock_069 prefetchsize 0;
alter tablespace ts_stock_070 prefetchsize 0;
alter tablespace ts_stock_071 prefetchsize 0;
alter tablespace ts_stock_072 prefetchsize 0;
alter tablespace ts_stock_073 prefetchsize 0;
alter tablespace ts_stock_074 prefetchsize 0;
alter tablespace ts_stock_075 prefetchsize 0;
alter tablespace ts_stock_076 prefetchsize 0;
alter tablespace ts_stock_077 prefetchsize 0;
alter tablespace ts_stock_078 prefetchsize 0;
alter tablespace ts_stock_079 prefetchsize 0;
alter tablespace ts_stock_080 prefetchsize 0;
alter tablespace ts_wh_001 prefetchsize 0;
alter tablespace ts_wh_002 prefetchsize 0;
alter tablespace ts_wh_003 prefetchsize 0;
alter tablespace ts_wh_004 prefetchsize 0;
alter tablespace ts_wh_005 prefetchsize 0;
alter tablespace ts_wh_006 prefetchsize 0;
alter tablespace ts_wh_007 prefetchsize 0;
alter tablespace ts_wh_008 prefetchsize 0;
alter tablespace ts_wh_009 prefetchsize 0;
alter tablespace ts_wh_010 prefetchsize 0;
alter tablespace ts_wh_011 prefetchsize 0;
alter tablespace ts_wh_012 prefetchsize 0;
alter tablespace ts_wh_013 prefetchsize 0;
alter tablespace ts_wh_014 prefetchsize 0;
alter tablespace ts_wh_015 prefetchsize 0;
alter tablespace ts_wh_016 prefetchsize 0;
alter tablespace ts_wh_017 prefetchsize 0;
alter tablespace ts_wh_018 prefetchsize 0;
alter tablespace ts_wh_019 prefetchsize 0;
alter tablespace ts_wh_020 prefetchsize 0;
alter tablespace ts_wh_021 prefetchsize 0;
alter tablespace ts_wh_022 prefetchsize 0;
```

```
alter tablespace ts_wh_023 prefetchsize 0;
alter tablespace ts_wh_024 prefetchsize 0;
alter tablespace ts_wh_025 prefetchsize 0;
alter tablespace ts_wh_026 prefetchsize 0;
alter tablespace ts_wh_027 prefetchsize 0;
alter tablespace ts_wh_028 prefetchsize 0;
alter tablespace ts_wh_029 prefetchsize 0;
alter tablespace ts_wh_030 prefetchsize 0;
alter tablespace ts_wh_031 prefetchsize 0;
alter tablespace ts_wh_032 prefetchsize 0;
alter tablespace ts_wh_033 prefetchsize 0;
alter tablespace ts_wh_034 prefetchsize 0;
alter tablespace ts_wh_035 prefetchsize 0;
alter tablespace ts_wh_036 prefetchsize 0;
alter tablespace ts_wh_037 prefetchsize 0;
alter tablespace ts_wh_038 prefetchsize 0;
alter tablespace ts_wh_039 prefetchsize 0;
alter tablespace ts_wh_040 prefetchsize 0;
alter tablespace ts_wh_041 prefetchsize 0;
alter tablespace ts_wh_042 prefetchsize 0;
alter tablespace ts_wh_043 prefetchsize 0;
alter tablespace ts_wh_044 prefetchsize 0;
alter tablespace ts_wh_045 prefetchsize 0;
alter tablespace ts_wh_046 prefetchsize 0;
alter tablespace ts_wh_047 prefetchsize 0;
alter tablespace ts_wh_048 prefetchsize 0;
alter tablespace ts_wh_049 prefetchsize 0;
alter tablespace ts_wh_050 prefetchsize 0;
alter tablespace ts_wh_051 prefetchsize 0;
alter tablespace ts_wh_052 prefetchsize 0;
alter tablespace ts_wh_053 prefetchsize 0;
alter tablespace ts_wh_054 prefetchsize 0;
alter tablespace ts_wh_055 prefetchsize 0;
alter tablespace ts_wh_056 prefetchsize 0;
alter tablespace ts_wh_057 prefetchsize 0;
alter tablespace ts_wh_058 prefetchsize 0;
alter tablespace ts_wh_059 prefetchsize 0;
alter tablespace ts_wh_060 prefetchsize 0;
alter tablespace ts_wh_061 prefetchsize 0;
alter tablespace ts_wh_062 prefetchsize 0;
alter tablespace ts_wh_063 prefetchsize 0;
alter tablespace ts_wh_064 prefetchsize 0;
alter tablespace ts_wh_065 prefetchsize 0;
alter tablespace ts_wh_066 prefetchsize 0;
alter tablespace ts_wh_067 prefetchsize 0;
alter tablespace ts_wh_068 prefetchsize 0;
alter tablespace ts_wh_069 prefetchsize 0;
alter tablespace ts_wh_070 prefetchsize 0;
alter tablespace ts_wh_071 prefetchsize 0;
alter tablespace ts_wh_072 prefetchsize 0;
alter tablespace ts_wh_073 prefetchsize 0;
alter tablespace ts_wh_074 prefetchsize 0;
alter tablespace ts_wh_075 prefetchsize 0;
alter tablespace ts_wh_076 prefetchsize 0;
alter tablespace ts_wh_077 prefetchsize 0;
alter tablespace ts_wh_078 prefetchsize 0;
alter tablespace ts_wh_079 prefetchsize 0;
alter tablespace ts_wh_080 prefetchsize 0;
connect reset;
```

DDL/ALTTBSP_PF_4096.ddl

```
connect to TPCC;
alter tablespace is_customer_001 prefetchsize 4096;
alter tablespace is_customer_002 prefetchsize 4096;
alter tablespace is_customer_003 prefetchsize 4096;
alter tablespace is_customer_004 prefetchsize 4096;
alter tablespace is_customer_005 prefetchsize 4096;
alter tablespace is_customer_006 prefetchsize 4096;
alter tablespace is_customer_007 prefetchsize 4096;
alter tablespace is_customer_008 prefetchsize 4096;
alter tablespace is_customer_009 prefetchsize 4096;
alter tablespace is_customer_010 prefetchsize 4096;
alter tablespace is_customer_011 prefetchsize 4096;
alter tablespace is_customer_012 prefetchsize 4096;
alter tablespace is_customer_013 prefetchsize 4096;
alter tablespace is_customer_014 prefetchsize 4096;
alter tablespace is_customer_015 prefetchsize 4096;
alter tablespace is_customer_016 prefetchsize 4096;
alter tablespace is_customer_017 prefetchsize 4096;
alter tablespace is_customer_018 prefetchsize 4096;
alter tablespace is_customer_019 prefetchsize 4096;
alter tablespace is_customer_020 prefetchsize 4096;
alter tablespace is_customer_021 prefetchsize 4096;
alter tablespace is_customer_022 prefetchsize 4096;
alter tablespace is_customer_023 prefetchsize 4096;
alter tablespace is_customer_024 prefetchsize 4096;
alter tablespace is_customer_025 prefetchsize 4096;
alter tablespace is_customer_026 prefetchsize 4096;
alter tablespace is_customer_027 prefetchsize 4096;
alter tablespace is_customer_028 prefetchsize 4096;
alter tablespace is_customer_029 prefetchsize 4096;
alter tablespace is_customer_030 prefetchsize 4096;
alter tablespace is_customer_031 prefetchsize 4096;
alter tablespace is_customer_032 prefetchsize 4096;
alter tablespace is_customer_033 prefetchsize 4096;
alter tablespace is_customer_034 prefetchsize 4096;
alter tablespace is_customer_035 prefetchsize 4096;
alter tablespace is_customer_036 prefetchsize 4096;
alter tablespace is_customer_037 prefetchsize 4096;
alter tablespace is_customer_038 prefetchsize 4096;
alter tablespace is_customer_039 prefetchsize 4096;
alter tablespace is_customer_040 prefetchsize 4096;
alter tablespace is_customer_041 prefetchsize 4096;
alter tablespace is_customer_042 prefetchsize 4096;
alter tablespace is_customer_043 prefetchsize 4096;
alter tablespace is_customer_044 prefetchsize 4096;
alter tablespace is_customer_045 prefetchsize 4096;
alter tablespace is_customer_046 prefetchsize 4096;
alter tablespace is_customer_047 prefetchsize 4096;
alter tablespace is_customer_048 prefetchsize 4096;
alter tablespace is_customer_049 prefetchsize 4096;
alter tablespace is_customer_050 prefetchsize 4096;
alter tablespace is_customer_051 prefetchsize 4096;
alter tablespace is_customer_052 prefetchsize 4096;
alter tablespace is_customer_053 prefetchsize 4096;
alter tablespace is_customer_054 prefetchsize 4096;
alter tablespace is_customer_055 prefetchsize 4096;
alter tablespace is_customer_056 prefetchsize 4096;
alter tablespace is_customer_057 prefetchsize 4096;
alter tablespace is_customer_058 prefetchsize 4096;
alter tablespace is_customer_059 prefetchsize 4096;
alter tablespace is_customer_060 prefetchsize 4096;
alter tablespace is_customer_061 prefetchsize 4096;
alter tablespace is_customer_062 prefetchsize 4096;
alter tablespace is_customer_063 prefetchsize 4096;
alter tablespace is_customer_064 prefetchsize 4096;
alter tablespace is_customer_065 prefetchsize 4096;
alter tablespace is_customer_066 prefetchsize 4096;
alter tablespace is_customer_067 prefetchsize 4096;
alter tablespace is_customer_068 prefetchsize 4096;
alter tablespace is_customer_069 prefetchsize 4096;
alter tablespace is_customer_070 prefetchsize 4096;
alter tablespace is_customer_071 prefetchsize 4096;
alter tablespace is_customer_072 prefetchsize 4096;
alter tablespace is_customer_073 prefetchsize 4096;
alter tablespace is_customer_074 prefetchsize 4096;
alter tablespace is_customer_075 prefetchsize 4096;
alter tablespace is_customer_076 prefetchsize 4096;
alter tablespace is_customer_077 prefetchsize 4096;
alter tablespace is_customer_078 prefetchsize 4096;
alter tablespace is_customer_079 prefetchsize 4096;
alter tablespace is_customer_080 prefetchsize 4096;
alter tablespace is_order_001 prefetchsize 4096;
alter tablespace is_order_002 prefetchsize 4096;
alter tablespace is_order_003 prefetchsize 4096;
alter tablespace is_order_004 prefetchsize 4096;
alter tablespace is_order_005 prefetchsize 4096;
alter tablespace is_order_006 prefetchsize 4096;
alter tablespace is_order_007 prefetchsize 4096;
alter tablespace is_order_008 prefetchsize 4096;
alter tablespace is_order_009 prefetchsize 4096;
alter tablespace is_order_010 prefetchsize 4096;
alter tablespace is_order_011 prefetchsize 4096;
alter tablespace is_order_012 prefetchsize 4096;
alter tablespace is_order_013 prefetchsize 4096;
alter tablespace is_order_014 prefetchsize 4096;
alter tablespace is_order_015 prefetchsize 4096;
alter tablespace is_order_016 prefetchsize 4096;
alter tablespace is_order_017 prefetchsize 4096;
alter tablespace is_order_018 prefetchsize 4096;
alter tablespace is_order_019 prefetchsize 4096;
alter tablespace is_order_020 prefetchsize 4096;
alter tablespace is_order_021 prefetchsize 4096;
alter tablespace is_order_022 prefetchsize 4096;
alter tablespace is_order_023 prefetchsize 4096;
alter tablespace is_order_024 prefetchsize 4096;
alter tablespace is_order_025 prefetchsize 4096;
alter tablespace is_order_026 prefetchsize 4096;
alter tablespace is_order_027 prefetchsize 4096;
alter tablespace is_order_028 prefetchsize 4096;
alter tablespace is_order_029 prefetchsize 4096;
alter tablespace is_order_030 prefetchsize 4096;
alter tablespace is_order_031 prefetchsize 4096;
alter tablespace is_order_032 prefetchsize 4096;
alter tablespace is_order_033 prefetchsize 4096;
alter tablespace is_order_034 prefetchsize 4096;
alter tablespace is_order_035 prefetchsize 4096;
alter tablespace is_order_036 prefetchsize 4096;
alter tablespace is_order_037 prefetchsize 4096;
alter tablespace is_order_038 prefetchsize 4096;
alter tablespace is_order_039 prefetchsize 4096;
alter tablespace is_order_040 prefetchsize 4096;
alter tablespace is_order_041 prefetchsize 4096;
alter tablespace is_order_042 prefetchsize 4096;
alter tablespace is_order_043 prefetchsize 4096;
alter tablespace is_order_044 prefetchsize 4096;
alter tablespace is_order_045 prefetchsize 4096;
alter tablespace is_order_046 prefetchsize 4096;
alter tablespace is_order_047 prefetchsize 4096;
alter tablespace is_order_048 prefetchsize 4096;
alter tablespace is_order_049 prefetchsize 4096;
alter tablespace is_order_050 prefetchsize 4096;
alter tablespace is_order_051 prefetchsize 4096;
alter tablespace is_order_052 prefetchsize 4096;
alter tablespace is_order_053 prefetchsize 4096;
alter tablespace is_order_054 prefetchsize 4096;
alter tablespace is_order_055 prefetchsize 4096;
alter tablespace is_order_056 prefetchsize 4096;
alter tablespace is_order_057 prefetchsize 4096;
alter tablespace is_order_058 prefetchsize 4096;
alter tablespace is_order_059 prefetchsize 4096;
alter tablespace is_order_060 prefetchsize 4096;
alter tablespace is_order_061 prefetchsize 4096;
alter tablespace is_order_062 prefetchsize 4096;
alter tablespace is_order_063 prefetchsize 4096;
alter tablespace is_order_064 prefetchsize 4096;
alter tablespace is_order_065 prefetchsize 4096;
alter tablespace is_order_066 prefetchsize 4096;
alter tablespace is_order_067 prefetchsize 4096;
```

```
alter tablespace is_order_068 prefetchsize 4096;
alter tablespace is_order_069 prefetchsize 4096;
alter tablespace is_order_070 prefetchsize 4096;
alter tablespace is_order_071 prefetchsize 4096;
alter tablespace is_order_072 prefetchsize 4096;
alter tablespace is_order_073 prefetchsize 4096;
alter tablespace is_order_074 prefetchsize 4096;
alter tablespace is_order_075 prefetchsize 4096;
alter tablespace is_order_076 prefetchsize 4096;
alter tablespace is_order_077 prefetchsize 4096;
alter tablespace is_order_078 prefetchsize 4096;
alter tablespace is_order_079 prefetchsize 4096;
alter tablespace is_order_080 prefetchsize 4096;
alter tablespace ts_customer_001 prefetchsize 4096;
alter tablespace ts_customer_002 prefetchsize 4096;
alter tablespace ts_customer_003 prefetchsize 4096;
alter tablespace ts_customer_004 prefetchsize 4096;
alter tablespace ts_customer_005 prefetchsize 4096;
alter tablespace ts_customer_006 prefetchsize 4096;
alter tablespace ts_customer_007 prefetchsize 4096;
alter tablespace ts_customer_008 prefetchsize 4096;
alter tablespace ts_customer_009 prefetchsize 4096;
alter tablespace ts_customer_010 prefetchsize 4096;
alter tablespace ts_customer_011 prefetchsize 4096;
alter tablespace ts_customer_012 prefetchsize 4096;
alter tablespace ts_customer_013 prefetchsize 4096;
alter tablespace ts_customer_014 prefetchsize 4096;
alter tablespace ts_customer_015 prefetchsize 4096;
alter tablespace ts_customer_016 prefetchsize 4096;
alter tablespace ts_customer_017 prefetchsize 4096;
alter tablespace ts_customer_018 prefetchsize 4096;
alter tablespace ts_customer_019 prefetchsize 4096;
alter tablespace ts_customer_020 prefetchsize 4096;
alter tablespace ts_customer_021 prefetchsize 4096;
alter tablespace ts_customer_022 prefetchsize 4096;
alter tablespace ts_customer_023 prefetchsize 4096;
alter tablespace ts_customer_024 prefetchsize 4096;
alter tablespace ts_customer_025 prefetchsize 4096;
alter tablespace ts_customer_026 prefetchsize 4096;
alter tablespace ts_customer_027 prefetchsize 4096;
alter tablespace ts_customer_028 prefetchsize 4096;
alter tablespace ts_customer_029 prefetchsize 4096;
alter tablespace ts_customer_030 prefetchsize 4096;
alter tablespace ts_customer_031 prefetchsize 4096;
alter tablespace ts_customer_032 prefetchsize 4096;
alter tablespace ts_customer_033 prefetchsize 4096;
alter tablespace ts_customer_034 prefetchsize 4096;
alter tablespace ts_customer_035 prefetchsize 4096;
alter tablespace ts_customer_036 prefetchsize 4096;
alter tablespace ts_customer_037 prefetchsize 4096;
alter tablespace ts_customer_038 prefetchsize 4096;
alter tablespace ts_customer_039 prefetchsize 4096;
alter tablespace ts_customer_040 prefetchsize 4096;
alter tablespace ts_customer_041 prefetchsize 4096;
alter tablespace ts_customer_042 prefetchsize 4096;
alter tablespace ts_customer_043 prefetchsize 4096;
alter tablespace ts_customer_044 prefetchsize 4096;
alter tablespace ts_customer_045 prefetchsize 4096;
alter tablespace ts_customer_046 prefetchsize 4096;
alter tablespace ts_customer_047 prefetchsize 4096;
alter tablespace ts_customer_048 prefetchsize 4096;
alter tablespace ts_customer_049 prefetchsize 4096;
alter tablespace ts_customer_050 prefetchsize 4096;
alter tablespace ts_customer_051 prefetchsize 4096;
alter tablespace ts_customer_052 prefetchsize 4096;
alter tablespace ts_customer_053 prefetchsize 4096;
alter tablespace ts_customer_054 prefetchsize 4096;
alter tablespace ts_customer_055 prefetchsize 4096;
alter tablespace ts_customer_056 prefetchsize 4096;
alter tablespace ts_customer_057 prefetchsize 4096;
alter tablespace ts_customer_058 prefetchsize 4096;
alter tablespace ts_customer_059 prefetchsize 4096;
alter tablespace ts_customer_060 prefetchsize 4096;
alter tablespace ts_customer_061 prefetchsize 4096;
alter tablespace ts_customer_062 prefetchsize 4096;
alter tablespace ts_customer_063 prefetchsize 4096;
alter tablespace ts_customer_064 prefetchsize 4096;
alter tablespace ts_customer_065 prefetchsize 4096;
alter tablespace ts_customer_066 prefetchsize 4096;
alter tablespace ts_customer_067 prefetchsize 4096;
alter tablespace ts_customer_068 prefetchsize 4096;
alter tablespace ts_customer_069 prefetchsize 4096;
alter tablespace ts_customer_070 prefetchsize 4096;
alter tablespace ts_customer_071 prefetchsize 4096;
alter tablespace ts_customer_072 prefetchsize 4096;
alter tablespace ts_customer_073 prefetchsize 4096;
alter tablespace ts_customer_074 prefetchsize 4096;
alter tablespace ts_customer_075 prefetchsize 4096;
alter tablespace ts_customer_076 prefetchsize 4096;
alter tablespace ts_customer_077 prefetchsize 4096;
alter tablespace ts_customer_078 prefetchsize 4096;
alter tablespace ts_customer_079 prefetchsize 4096;
alter tablespace ts_customer_080 prefetchsize 4096;
alter tablespace ts_dis_001 prefetchsize 4096;
alter tablespace ts_dis_002 prefetchsize 4096;
alter tablespace ts_dis_003 prefetchsize 4096;
alter tablespace ts_dis_004 prefetchsize 4096;
alter tablespace ts_dis_005 prefetchsize 4096;
alter tablespace ts_dis_006 prefetchsize 4096;
alter tablespace ts_dis_007 prefetchsize 4096;
alter tablespace ts_dis_008 prefetchsize 4096;
alter tablespace ts_dis_009 prefetchsize 4096;
alter tablespace ts_dis_010 prefetchsize 4096;
alter tablespace ts_dis_011 prefetchsize 4096;
alter tablespace ts_dis_012 prefetchsize 4096;
alter tablespace ts_dis_013 prefetchsize 4096;
alter tablespace ts_dis_014 prefetchsize 4096;
alter tablespace ts_dis_015 prefetchsize 4096;
alter tablespace ts_dis_016 prefetchsize 4096;
alter tablespace ts_dis_017 prefetchsize 4096;
alter tablespace ts_dis_018 prefetchsize 4096;
alter tablespace ts_dis_019 prefetchsize 4096;
alter tablespace ts_dis_020 prefetchsize 4096;
alter tablespace ts_dis_021 prefetchsize 4096;
alter tablespace ts_dis_022 prefetchsize 4096;
alter tablespace ts_dis_023 prefetchsize 4096;
alter tablespace ts_dis_024 prefetchsize 4096;
alter tablespace ts_dis_025 prefetchsize 4096;
alter tablespace ts_dis_026 prefetchsize 4096;
alter tablespace ts_dis_027 prefetchsize 4096;
alter tablespace ts_dis_028 prefetchsize 4096;
alter tablespace ts_dis_029 prefetchsize 4096;
alter tablespace ts_dis_030 prefetchsize 4096;
alter tablespace ts_dis_031 prefetchsize 4096;
alter tablespace ts_dis_032 prefetchsize 4096;
alter tablespace ts_dis_033 prefetchsize 4096;
alter tablespace ts_dis_034 prefetchsize 4096;
alter tablespace ts_dis_035 prefetchsize 4096;
alter tablespace ts_dis_036 prefetchsize 4096;
alter tablespace ts_dis_037 prefetchsize 4096;
alter tablespace ts_dis_038 prefetchsize 4096;
alter tablespace ts_dis_039 prefetchsize 4096;
alter tablespace ts_dis_040 prefetchsize 4096;
alter tablespace ts_dis_041 prefetchsize 4096;
alter tablespace ts_dis_042 prefetchsize 4096;
alter tablespace ts_dis_043 prefetchsize 4096;
alter tablespace ts_dis_044 prefetchsize 4096;
alter tablespace ts_dis_045 prefetchsize 4096;
alter tablespace ts_dis_046 prefetchsize 4096;
alter tablespace ts_dis_047 prefetchsize 4096;
alter tablespace ts_dis_048 prefetchsize 4096;
alter tablespace ts_dis_049 prefetchsize 4096;
alter tablespace ts_dis_050 prefetchsize 4096;
alter tablespace ts_dis_051 prefetchsize 4096;
alter tablespace ts_dis_052 prefetchsize 4096;
alter tablespace ts_dis_053 prefetchsize 4096;
alter tablespace ts_dis_054 prefetchsize 4096;
alter tablespace ts_dis_055 prefetchsize 4096;
alter tablespace ts_dis_056 prefetchsize 4096;
alter tablespace ts_dis_057 prefetchsize 4096;
alter tablespace ts_dis_058 prefetchsize 4096;
alter tablespace ts_dis_059 prefetchsize 4096;
alter tablespace ts_dis_060 prefetchsize 4096;
alter tablespace ts_dis_061 prefetchsize 4096;
alter tablespace ts_dis_062 prefetchsize 4096;
alter tablespace ts_dis_063 prefetchsize 4096;
alter tablespace ts_dis_064 prefetchsize 4096;
alter tablespace ts_dis_065 prefetchsize 4096;
alter tablespace ts_dis_066 prefetchsize 4096;
alter tablespace ts_dis_067 prefetchsize 4096;
alter tablespace ts_dis_068 prefetchsize 4096;
alter tablespace ts_dis_069 prefetchsize 4096;
alter tablespace ts_dis_070 prefetchsize 4096;
alter tablespace ts_dis_071 prefetchsize 4096;
alter tablespace ts_dis_072 prefetchsize 4096;
alter tablespace ts_dis_073 prefetchsize 4096;
alter tablespace ts_dis_074 prefetchsize 4096;
alter tablespace ts_dis_075 prefetchsize 4096;
alter tablespace ts_dis_076 prefetchsize 4096;
alter tablespace ts_dis_077 prefetchsize 4096;
alter tablespace ts_dis_078 prefetchsize 4096;
alter tablespace ts_dis_079 prefetchsize 4096;
alter tablespace ts_dis_080 prefetchsize 4096;
alter tablespace ts_history_001 prefetchsize 4096;
alter tablespace ts_history_002 prefetchsize 4096;
alter tablespace ts_history_003 prefetchsize 4096;
alter tablespace ts_history_004 prefetchsize 4096;
alter tablespace ts_history_005 prefetchsize 4096;
alter tablespace ts_history_006 prefetchsize 4096;
alter tablespace ts_history_007 prefetchsize 4096;
alter tablespace ts_history_008 prefetchsize 4096;
alter tablespace ts_history_009 prefetchsize 4096;
alter tablespace ts_history_010 prefetchsize 4096;
alter tablespace ts_history_011 prefetchsize 4096;
alter tablespace ts_history_012 prefetchsize 4096;
alter tablespace ts_history_013 prefetchsize 4096;
alter tablespace ts_history_014 prefetchsize 4096;
alter tablespace ts_history_015 prefetchsize 4096;
alter tablespace ts_history_016 prefetchsize 4096;
alter tablespace ts_history_017 prefetchsize 4096;
alter tablespace ts_history_018 prefetchsize 4096;
alter tablespace ts_history_019 prefetchsize 4096;
alter tablespace ts_history_020 prefetchsize 4096;
alter tablespace ts_history_021 prefetchsize 4096;
alter tablespace ts_history_022 prefetchsize 4096;
alter tablespace ts_history_023 prefetchsize 4096;
alter tablespace ts_history_024 prefetchsize 4096;
alter tablespace ts_history_025 prefetchsize 4096;
alter tablespace ts_history_026 prefetchsize 4096;
alter tablespace ts_history_027 prefetchsize 4096;
alter tablespace ts_history_028 prefetchsize 4096;
alter tablespace ts_history_029 prefetchsize 4096;
alter tablespace ts_history_030 prefetchsize 4096;
alter tablespace ts_history_031 prefetchsize 4096;
alter tablespace ts_history_032 prefetchsize 4096;
alter tablespace ts_history_033 prefetchsize 4096;
alter tablespace ts_history_034 prefetchsize 4096;
alter tablespace ts_history_035 prefetchsize 4096;
alter tablespace ts_history_036 prefetchsize 4096;
alter tablespace ts_history_037 prefetchsize 4096;
```

```
alter tablespace ts_history_038 prefetchsize 4096;
alter tablespace ts_history_039 prefetchsize 4096;
alter tablespace ts_history_040 prefetchsize 4096;
alter tablespace ts_history_041 prefetchsize 4096;
alter tablespace ts_history_042 prefetchsize 4096;
alter tablespace ts_history_043 prefetchsize 4096;
alter tablespace ts_history_044 prefetchsize 4096;
alter tablespace ts_history_045 prefetchsize 4096;
alter tablespace ts_history_046 prefetchsize 4096;
alter tablespace ts_history_047 prefetchsize 4096;
alter tablespace ts_history_048 prefetchsize 4096;
alter tablespace ts_history_049 prefetchsize 4096;
alter tablespace ts_history_050 prefetchsize 4096;
alter tablespace ts_history_051 prefetchsize 4096;
alter tablespace ts_history_052 prefetchsize 4096;
alter tablespace ts_history_053 prefetchsize 4096;
alter tablespace ts_history_054 prefetchsize 4096;
alter tablespace ts_history_055 prefetchsize 4096;
alter tablespace ts_history_056 prefetchsize 4096;
alter tablespace ts_history_057 prefetchsize 4096;
alter tablespace ts_history_058 prefetchsize 4096;
alter tablespace ts_history_059 prefetchsize 4096;
alter tablespace ts_history_060 prefetchsize 4096;
alter tablespace ts_history_061 prefetchsize 4096;
alter tablespace ts_history_062 prefetchsize 4096;
alter tablespace ts_history_063 prefetchsize 4096;
alter tablespace ts_history_064 prefetchsize 4096;
alter tablespace ts_history_065 prefetchsize 4096;
alter tablespace ts_history_066 prefetchsize 4096;
alter tablespace ts_history_067 prefetchsize 4096;
alter tablespace ts_history_068 prefetchsize 4096;
alter tablespace ts_history_069 prefetchsize 4096;
alter tablespace ts_history_070 prefetchsize 4096;
alter tablespace ts_history_071 prefetchsize 4096;
alter tablespace ts_history_072 prefetchsize 4096;
alter tablespace ts_history_073 prefetchsize 4096;
alter tablespace ts_history_074 prefetchsize 4096;
alter tablespace ts_history_075 prefetchsize 4096;
alter tablespace ts_history_076 prefetchsize 4096;
alter tablespace ts_history_077 prefetchsize 4096;
alter tablespace ts_history_078 prefetchsize 4096;
alter tablespace ts_history_079 prefetchsize 4096;
alter tablespace ts_history_080 prefetchsize 4096;
alter tablespace ts_item prefetchsize 4096;
alter tablespace ts_newordA_001 prefetchsize 4096;
alter tablespace ts_newordA_002 prefetchsize 4096;
alter tablespace ts_newordA_003 prefetchsize 4096;
alter tablespace ts_newordA_004 prefetchsize 4096;
alter tablespace ts_newordA_005 prefetchsize 4096;
alter tablespace ts_newordA_006 prefetchsize 4096;
alter tablespace ts_newordA_007 prefetchsize 4096;
alter tablespace ts_newordA_008 prefetchsize 4096;
alter tablespace ts_newordA_009 prefetchsize 4096;
alter tablespace ts_newordA_010 prefetchsize 4096;
alter tablespace ts_newordA_011 prefetchsize 4096;
alter tablespace ts_newordA_012 prefetchsize 4096;
alter tablespace ts_newordA_013 prefetchsize 4096;
alter tablespace ts_newordA_014 prefetchsize 4096;
alter tablespace ts_newordA_015 prefetchsize 4096;
alter tablespace ts_newordA_016 prefetchsize 4096;
alter tablespace ts_newordA_017 prefetchsize 4096;
alter tablespace ts_newordA_018 prefetchsize 4096;
alter tablespace ts_newordA_019 prefetchsize 4096;
alter tablespace ts_newordA_020 prefetchsize 4096;
alter tablespace ts_newordA_021 prefetchsize 4096;
alter tablespace ts_newordA_022 prefetchsize 4096;
alter tablespace ts_newordA_023 prefetchsize 4096;
alter tablespace ts_newordA_024 prefetchsize 4096;
alter tablespace ts_newordA_025 prefetchsize 4096;
alter tablespace ts_newordA_026 prefetchsize 4096;
alter tablespace ts_newordA_027 prefetchsize 4096;
alter tablespace ts_newordA_028 prefetchsize 4096;
alter tablespace ts_newordA_029 prefetchsize 4096;
alter tablespace ts_newordA_030 prefetchsize 4096;
alter tablespace ts_newordA_031 prefetchsize 4096;
alter tablespace ts_newordA_032 prefetchsize 4096;
alter tablespace ts_newordA_033 prefetchsize 4096;
alter tablespace ts_newordA_034 prefetchsize 4096;
alter tablespace ts_newordA_035 prefetchsize 4096;
alter tablespace ts_newordA_036 prefetchsize 4096;
alter tablespace ts_newordA_037 prefetchsize 4096;
alter tablespace ts_newordA_038 prefetchsize 4096;
alter tablespace ts_newordA_039 prefetchsize 4096;
alter tablespace ts_newordA_040 prefetchsize 4096;
alter tablespace ts_newordA_041 prefetchsize 4096;
alter tablespace ts_newordA_042 prefetchsize 4096;
alter tablespace ts_newordA_043 prefetchsize 4096;
alter tablespace ts_newordA_044 prefetchsize 4096;
alter tablespace ts_newordA_045 prefetchsize 4096;
alter tablespace ts_newordA_046 prefetchsize 4096;
alter tablespace ts_newordA_047 prefetchsize 4096;
alter tablespace ts_newordA_048 prefetchsize 4096;
alter tablespace ts_newordA_049 prefetchsize 4096;
alter tablespace ts_newordA_050 prefetchsize 4096;
alter tablespace ts_newordA_051 prefetchsize 4096;
alter tablespace ts_newordA_052 prefetchsize 4096;
alter tablespace ts_newordA_053 prefetchsize 4096;
alter tablespace ts_newordA_054 prefetchsize 4096;
alter tablespace ts_newordA_055 prefetchsize 4096;
alter tablespace ts_newordA_056 prefetchsize 4096;
alter tablespace ts_newordA_057 prefetchsize 4096;
alter tablespace ts_newordA_058 prefetchsize 4096;
alter tablespace ts_newordA_059 prefetchsize 4096;
alter tablespace ts_newordA_060 prefetchsize 4096;
alter tablespace ts_newordA_061 prefetchsize 4096;
alter tablespace ts_newordA_062 prefetchsize 4096;
alter tablespace ts_newordA_063 prefetchsize 4096;
alter tablespace ts_newordA_064 prefetchsize 4096;
alter tablespace ts_newordA_065 prefetchsize 4096;
alter tablespace ts_newordA_066 prefetchsize 4096;
alter tablespace ts_newordA_067 prefetchsize 4096;
alter tablespace ts_newordA_068 prefetchsize 4096;
alter tablespace ts_newordA_069 prefetchsize 4096;
alter tablespace ts_newordA_070 prefetchsize 4096;
alter tablespace ts_newordA_071 prefetchsize 4096;
alter tablespace ts_newordA_072 prefetchsize 4096;
alter tablespace ts_newordA_073 prefetchsize 4096;
alter tablespace ts_newordA_074 prefetchsize 4096;
alter tablespace ts_newordA_075 prefetchsize 4096;
alter tablespace ts_newordA_076 prefetchsize 4096;
alter tablespace ts_newordA_077 prefetchsize 4096;
alter tablespace ts_newordA_078 prefetchsize 4096;
alter tablespace ts_newordA_079 prefetchsize 4096;
alter tablespace ts_newordA_080 prefetchsize 4096;
alter tablespace ts_newordB_001 prefetchsize 4096;
alter tablespace ts_newordB_002 prefetchsize 4096;
alter tablespace ts_newordB_003 prefetchsize 4096;
alter tablespace ts_newordB_004 prefetchsize 4096;
alter tablespace ts_newordB_005 prefetchsize 4096;
alter tablespace ts_newordB_006 prefetchsize 4096;
alter tablespace ts_newordB_007 prefetchsize 4096;
alter tablespace ts_newordB_008 prefetchsize 4096;
alter tablespace ts_newordB_009 prefetchsize 4096;
alter tablespace ts_newordB_010 prefetchsize 4096;
alter tablespace ts_newordB_011 prefetchsize 4096;
alter tablespace ts_newordB_012 prefetchsize 4096;
alter tablespace ts_newordB_013 prefetchsize 4096;
alter tablespace ts_newordB_014 prefetchsize 4096;
alter tablespace ts_newordB_015 prefetchsize 4096;
alter tablespace ts_newordB_016 prefetchsize 4096;
alter tablespace ts_newordB_017 prefetchsize 4096;
alter tablespace ts_newordB_018 prefetchsize 4096;
alter tablespace ts_newordB_019 prefetchsize 4096;
alter tablespace ts_newordB_020 prefetchsize 4096;
alter tablespace ts_newordB_021 prefetchsize 4096;
alter tablespace ts_newordB_022 prefetchsize 4096;
alter tablespace ts_newordB_023 prefetchsize 4096;
alter tablespace ts_newordB_024 prefetchsize 4096;
alter tablespace ts_newordB_025 prefetchsize 4096;
alter tablespace ts_newordB_026 prefetchsize 4096;
alter tablespace ts_newordB_027 prefetchsize 4096;
alter tablespace ts_newordB_028 prefetchsize 4096;
alter tablespace ts_newordB_029 prefetchsize 4096;
alter tablespace ts_newordB_030 prefetchsize 4096;
alter tablespace ts_newordB_031 prefetchsize 4096;
alter tablespace ts_newordB_032 prefetchsize 4096;
alter tablespace ts_newordB_033 prefetchsize 4096;
alter tablespace ts_newordB_034 prefetchsize 4096;
alter tablespace ts_newordB_035 prefetchsize 4096;
alter tablespace ts_newordB_036 prefetchsize 4096;
alter tablespace ts_newordB_037 prefetchsize 4096;
alter tablespace ts_newordB_038 prefetchsize 4096;
alter tablespace ts_newordB_039 prefetchsize 4096;
alter tablespace ts_newordB_040 prefetchsize 4096;
alter tablespace ts_newordB_041 prefetchsize 4096;
alter tablespace ts_newordB_042 prefetchsize 4096;
alter tablespace ts_newordB_043 prefetchsize 4096;
alter tablespace ts_newordB_044 prefetchsize 4096;
alter tablespace ts_newordB_045 prefetchsize 4096;
alter tablespace ts_newordB_046 prefetchsize 4096;
alter tablespace ts_newordB_047 prefetchsize 4096;
alter tablespace ts_newordB_048 prefetchsize 4096;
alter tablespace ts_newordB_049 prefetchsize 4096;
alter tablespace ts_newordB_050 prefetchsize 4096;
alter tablespace ts_newordB_051 prefetchsize 4096;
alter tablespace ts_newordB_052 prefetchsize 4096;
alter tablespace ts_newordB_053 prefetchsize 4096;
alter tablespace ts_newordB_054 prefetchsize 4096;
alter tablespace ts_newordB_055 prefetchsize 4096;
alter tablespace ts_newordB_056 prefetchsize 4096;
alter tablespace ts_newordB_057 prefetchsize 4096;
alter tablespace ts_newordB_058 prefetchsize 4096;
alter tablespace ts_newordB_059 prefetchsize 4096;
alter tablespace ts_newordB_060 prefetchsize 4096;
alter tablespace ts_newordB_061 prefetchsize 4096;
alter tablespace ts_newordB_062 prefetchsize 4096;
alter tablespace ts_newordB_063 prefetchsize 4096;
alter tablespace ts_newordB_064 prefetchsize 4096;
alter tablespace ts_newordB_065 prefetchsize 4096;
alter tablespace ts_newordB_066 prefetchsize 4096;
alter tablespace ts_newordB_067 prefetchsize 4096;
alter tablespace ts_newordB_068 prefetchsize 4096;
alter tablespace ts_newordB_069 prefetchsize 4096;
alter tablespace ts_newordB_070 prefetchsize 4096;
alter tablespace ts_newordB_071 prefetchsize 4096;
alter tablespace ts_newordB_072 prefetchsize 4096;
alter tablespace ts_newordB_073 prefetchsize 4096;
alter tablespace ts_newordB_074 prefetchsize 4096;
alter tablespace ts_newordB_075 prefetchsize 4096;
alter tablespace ts_newordB_076 prefetchsize 4096;
alter tablespace ts_newordB_077 prefetchsize 4096;
alter tablespace ts_newordB_078 prefetchsize 4096;
alter tablespace ts_newordB_079 prefetchsize 4096;
alter tablespace ts_newordB_080 prefetchsize 4096;
alter tablespace ts_order_001 prefetchsize 4096;
alter tablespace ts_order_002 prefetchsize 4096;
alter tablespace ts_order_003 prefetchsize 4096;
alter tablespace ts_order_004 prefetchsize 4096;
alter tablespace ts_order_005 prefetchsize 4096;
alter tablespace ts_order_006 prefetchsize 4096;
```

```
alter tablespace ts_order_007 prefetchsize 4096;
alter tablespace ts_order_008 prefetchsize 4096;
alter tablespace ts_order_009 prefetchsize 4096;
alter tablespace ts_order_010 prefetchsize 4096;
alter tablespace ts_order_011 prefetchsize 4096;
alter tablespace ts_order_012 prefetchsize 4096;
alter tablespace ts_order_013 prefetchsize 4096;
alter tablespace ts_order_014 prefetchsize 4096;
alter tablespace ts_order_015 prefetchsize 4096;
alter tablespace ts_order_016 prefetchsize 4096;
alter tablespace ts_order_017 prefetchsize 4096;
alter tablespace ts_order_018 prefetchsize 4096;
alter tablespace ts_order_019 prefetchsize 4096;
alter tablespace ts_order_020 prefetchsize 4096;
alter tablespace ts_order_021 prefetchsize 4096;
alter tablespace ts_order_022 prefetchsize 4096;
alter tablespace ts_order_023 prefetchsize 4096;
alter tablespace ts_order_024 prefetchsize 4096;
alter tablespace ts_order_025 prefetchsize 4096;
alter tablespace ts_order_026 prefetchsize 4096;
alter tablespace ts_order_027 prefetchsize 4096;
alter tablespace ts_order_028 prefetchsize 4096;
alter tablespace ts_order_029 prefetchsize 4096;
alter tablespace ts_order_030 prefetchsize 4096;
alter tablespace ts_order_031 prefetchsize 4096;
alter tablespace ts_order_032 prefetchsize 4096;
alter tablespace ts_order_033 prefetchsize 4096;
alter tablespace ts_order_034 prefetchsize 4096;
alter tablespace ts_order_035 prefetchsize 4096;
alter tablespace ts_order_036 prefetchsize 4096;
alter tablespace ts_order_037 prefetchsize 4096;
alter tablespace ts_order_038 prefetchsize 4096;
alter tablespace ts_order_039 prefetchsize 4096;
alter tablespace ts_order_040 prefetchsize 4096;
alter tablespace ts_order_041 prefetchsize 4096;
alter tablespace ts_order_042 prefetchsize 4096;
alter tablespace ts_order_043 prefetchsize 4096;
alter tablespace ts_order_044 prefetchsize 4096;
alter tablespace ts_order_045 prefetchsize 4096;
alter tablespace ts_order_046 prefetchsize 4096;
alter tablespace ts_order_047 prefetchsize 4096;
alter tablespace ts_order_048 prefetchsize 4096;
alter tablespace ts_order_049 prefetchsize 4096;
alter tablespace ts_order_050 prefetchsize 4096;
alter tablespace ts_order_051 prefetchsize 4096;
alter tablespace ts_order_052 prefetchsize 4096;
alter tablespace ts_order_053 prefetchsize 4096;
alter tablespace ts_order_054 prefetchsize 4096;
alter tablespace ts_order_055 prefetchsize 4096;
alter tablespace ts_order_056 prefetchsize 4096;
alter tablespace ts_order_057 prefetchsize 4096;
alter tablespace ts_order_058 prefetchsize 4096;
alter tablespace ts_order_059 prefetchsize 4096;
alter tablespace ts_order_060 prefetchsize 4096;
alter tablespace ts_order_061 prefetchsize 4096;
alter tablespace ts_order_062 prefetchsize 4096;
alter tablespace ts_order_063 prefetchsize 4096;
alter tablespace ts_order_064 prefetchsize 4096;
alter tablespace ts_order_065 prefetchsize 4096;
alter tablespace ts_order_066 prefetchsize 4096;
alter tablespace ts_order_067 prefetchsize 4096;
alter tablespace ts_order_068 prefetchsize 4096;
alter tablespace ts_order_069 prefetchsize 4096;
alter tablespace ts_order_070 prefetchsize 4096;
alter tablespace ts_order_071 prefetchsize 4096;
alter tablespace ts_order_072 prefetchsize 4096;
alter tablespace ts_order_073 prefetchsize 4096;
alter tablespace ts_order_074 prefetchsize 4096;
alter tablespace ts_order_075 prefetchsize 4096;
alter tablespace ts_order_076 prefetchsize 4096;
alter tablespace ts_order_077 prefetchsize 4096;
alter tablespace ts_order_078 prefetchsize 4096;
alter tablespace ts_order_079 prefetchsize 4096;
alter tablespace ts_order_080 prefetchsize 4096;
alter tablespace ts_orderline_001 prefetchsize 4096;
alter tablespace ts_orderline_002 prefetchsize 4096;
alter tablespace ts_orderline_003 prefetchsize 4096;
alter tablespace ts_orderline_004 prefetchsize 4096;
alter tablespace ts_orderline_005 prefetchsize 4096;
alter tablespace ts_orderline_006 prefetchsize 4096;
alter tablespace ts_orderline_007 prefetchsize 4096;
alter tablespace ts_orderline_008 prefetchsize 4096;
alter tablespace ts_orderline_009 prefetchsize 4096;
alter tablespace ts_orderline_010 prefetchsize 4096;
alter tablespace ts_orderline_011 prefetchsize 4096;
alter tablespace ts_orderline_012 prefetchsize 4096;
alter tablespace ts_orderline_013 prefetchsize 4096;
alter tablespace ts_orderline_014 prefetchsize 4096;
alter tablespace ts_orderline_015 prefetchsize 4096;
alter tablespace ts_orderline_016 prefetchsize 4096;
alter tablespace ts_orderline_017 prefetchsize 4096;
alter tablespace ts_orderline_018 prefetchsize 4096;
alter tablespace ts_orderline_019 prefetchsize 4096;
alter tablespace ts_orderline_020 prefetchsize 4096;
alter tablespace ts_orderline_021 prefetchsize 4096;
alter tablespace ts_orderline_022 prefetchsize 4096;
alter tablespace ts_orderline_023 prefetchsize 4096;
alter tablespace ts_orderline_024 prefetchsize 4096;
alter tablespace ts_orderline_025 prefetchsize 4096;
alter tablespace ts_orderline_026 prefetchsize 4096;
alter tablespace ts_orderline_027 prefetchsize 4096;
alter tablespace ts_orderline_028 prefetchsize 4096;
alter tablespace ts_orderline_029 prefetchsize 4096;
alter tablespace ts_orderline_030 prefetchsize 4096;
alter tablespace ts_orderline_031 prefetchsize 4096;
alter tablespace ts_orderline_032 prefetchsize 4096;
alter tablespace ts_orderline_033 prefetchsize 4096;
alter tablespace ts_orderline_034 prefetchsize 4096;
alter tablespace ts_orderline_035 prefetchsize 4096;
alter tablespace ts_orderline_036 prefetchsize 4096;
alter tablespace ts_orderline_037 prefetchsize 4096;
alter tablespace ts_orderline_038 prefetchsize 4096;
alter tablespace ts_orderline_039 prefetchsize 4096;
alter tablespace ts_orderline_040 prefetchsize 4096;
alter tablespace ts_orderline_041 prefetchsize 4096;
alter tablespace ts_orderline_042 prefetchsize 4096;
alter tablespace ts_orderline_043 prefetchsize 4096;
alter tablespace ts_orderline_044 prefetchsize 4096;
alter tablespace ts_orderline_045 prefetchsize 4096;
alter tablespace ts_orderline_046 prefetchsize 4096;
alter tablespace ts_orderline_047 prefetchsize 4096;
alter tablespace ts_orderline_048 prefetchsize 4096;
alter tablespace ts_orderline_049 prefetchsize 4096;
alter tablespace ts_orderline_050 prefetchsize 4096;
alter tablespace ts_orderline_051 prefetchsize 4096;
alter tablespace ts_orderline_052 prefetchsize 4096;
alter tablespace ts_orderline_053 prefetchsize 4096;
alter tablespace ts_orderline_054 prefetchsize 4096;
alter tablespace ts_orderline_055 prefetchsize 4096;
alter tablespace ts_orderline_056 prefetchsize 4096;
alter tablespace ts_orderline_057 prefetchsize 4096;
alter tablespace ts_orderline_058 prefetchsize 4096;
alter tablespace ts_orderline_059 prefetchsize 4096;
alter tablespace ts_orderline_060 prefetchsize 4096;
alter tablespace ts_orderline_061 prefetchsize 4096;
alter tablespace ts_orderline_062 prefetchsize 4096;
alter tablespace ts_orderline_063 prefetchsize 4096;
alter tablespace ts_orderline_064 prefetchsize 4096;
alter tablespace ts_orderline_065 prefetchsize 4096;
alter tablespace ts_orderline_066 prefetchsize 4096;
alter tablespace ts_orderline_067 prefetchsize 4096;
alter tablespace ts_orderline_068 prefetchsize 4096;
alter tablespace ts_orderline_069 prefetchsize 4096;
alter tablespace ts_orderline_070 prefetchsize 4096;
alter tablespace ts_orderline_071 prefetchsize 4096;
alter tablespace ts_orderline_072 prefetchsize 4096;
alter tablespace ts_orderline_073 prefetchsize 4096;
alter tablespace ts_orderline_074 prefetchsize 4096;
alter tablespace ts_orderline_075 prefetchsize 4096;
alter tablespace ts_orderline_076 prefetchsize 4096;
alter tablespace ts_orderline_077 prefetchsize 4096;
alter tablespace ts_orderline_078 prefetchsize 4096;
alter tablespace ts_orderline_079 prefetchsize 4096;
alter tablespace ts_orderline_080 prefetchsize 4096;
alter tablespace ts_stock_001 prefetchsize 4096;
alter tablespace ts_stock_002 prefetchsize 4096;
alter tablespace ts_stock_003 prefetchsize 4096;
alter tablespace ts_stock_004 prefetchsize 4096;
alter tablespace ts_stock_005 prefetchsize 4096;
alter tablespace ts_stock_006 prefetchsize 4096;
alter tablespace ts_stock_007 prefetchsize 4096;
alter tablespace ts_stock_008 prefetchsize 4096;
alter tablespace ts_stock_009 prefetchsize 4096;
alter tablespace ts_stock_010 prefetchsize 4096;
alter tablespace ts_stock_011 prefetchsize 4096;
alter tablespace ts_stock_012 prefetchsize 4096;
alter tablespace ts_stock_013 prefetchsize 4096;
alter tablespace ts_stock_014 prefetchsize 4096;
alter tablespace ts_stock_015 prefetchsize 4096;
alter tablespace ts_stock_016 prefetchsize 4096;
alter tablespace ts_stock_017 prefetchsize 4096;
alter tablespace ts_stock_018 prefetchsize 4096;
alter tablespace ts_stock_019 prefetchsize 4096;
alter tablespace ts_stock_020 prefetchsize 4096;
alter tablespace ts_stock_021 prefetchsize 4096;
alter tablespace ts_stock_022 prefetchsize 4096;
alter tablespace ts_stock_023 prefetchsize 4096;
alter tablespace ts_stock_024 prefetchsize 4096;
alter tablespace ts_stock_025 prefetchsize 4096;
alter tablespace ts_stock_026 prefetchsize 4096;
alter tablespace ts_stock_027 prefetchsize 4096;
alter tablespace ts_stock_028 prefetchsize 4096;
alter tablespace ts_stock_029 prefetchsize 4096;
alter tablespace ts_stock_030 prefetchsize 4096;
alter tablespace ts_stock_031 prefetchsize 4096;
alter tablespace ts_stock_032 prefetchsize 4096;
alter tablespace ts_stock_033 prefetchsize 4096;
alter tablespace ts_stock_034 prefetchsize 4096;
alter tablespace ts_stock_035 prefetchsize 4096;
alter tablespace ts_stock_036 prefetchsize 4096;
alter tablespace ts_stock_037 prefetchsize 4096;
alter tablespace ts_stock_038 prefetchsize 4096;
alter tablespace ts_stock_039 prefetchsize 4096;
alter tablespace ts_stock_040 prefetchsize 4096;
alter tablespace ts_stock_041 prefetchsize 4096;
alter tablespace ts_stock_042 prefetchsize 4096;
alter tablespace ts_stock_043 prefetchsize 4096;
alter tablespace ts_stock_044 prefetchsize 4096;
alter tablespace ts_stock_045 prefetchsize 4096;
alter tablespace ts_stock_046 prefetchsize 4096;
alter tablespace ts_stock_047 prefetchsize 4096;
alter tablespace ts_stock_048 prefetchsize 4096;
alter tablespace ts_stock_049 prefetchsize 4096;
alter tablespace ts_stock_050 prefetchsize 4096;
alter tablespace ts_stock_051 prefetchsize 4096;
alter tablespace ts_stock_052 prefetchsize 4096;
alter tablespace ts_stock_053 prefetchsize 4096;
alter tablespace ts_stock_054 prefetchsize 4096;
alter tablespace ts_stock_055 prefetchsize 4096;
alter tablespace ts_stock_056 prefetchsize 4096;
```

```
alter tablespace ts_stock_057 prefetchsize 4096;
alter tablespace ts_stock_058 prefetchsize 4096;
alter tablespace ts_stock_059 prefetchsize 4096;
alter tablespace ts_stock_060 prefetchsize 4096;
alter tablespace ts_stock_061 prefetchsize 4096;
alter tablespace ts_stock_062 prefetchsize 4096;
alter tablespace ts_stock_063 prefetchsize 4096;
alter tablespace ts_stock_064 prefetchsize 4096;
alter tablespace ts_stock_065 prefetchsize 4096;
alter tablespace ts_stock_066 prefetchsize 4096;
alter tablespace ts_stock_067 prefetchsize 4096;
alter tablespace ts_stock_068 prefetchsize 4096;
alter tablespace ts_stock_069 prefetchsize 4096;
alter tablespace ts_stock_070 prefetchsize 4096;
alter tablespace ts_stock_071 prefetchsize 4096;
alter tablespace ts_stock_072 prefetchsize 4096;
alter tablespace ts_stock_073 prefetchsize 4096;
alter tablespace ts_stock_074 prefetchsize 4096;
alter tablespace ts_stock_075 prefetchsize 4096;
alter tablespace ts_stock_076 prefetchsize 4096;
alter tablespace ts_stock_077 prefetchsize 4096;
alter tablespace ts_stock_078 prefetchsize 4096;
alter tablespace ts_stock_079 prefetchsize 4096;
alter tablespace ts_stock_080 prefetchsize 4096;
alter tablespace ts_wh_001 prefetchsize 4096;
alter tablespace ts_wh_002 prefetchsize 4096;
alter tablespace ts_wh_003 prefetchsize 4096;
alter tablespace ts_wh_004 prefetchsize 4096;
alter tablespace ts_wh_005 prefetchsize 4096;
alter tablespace ts_wh_006 prefetchsize 4096;
alter tablespace ts_wh_007 prefetchsize 4096;
alter tablespace ts_wh_008 prefetchsize 4096;
alter tablespace ts_wh_009 prefetchsize 4096;
alter tablespace ts_wh_010 prefetchsize 4096;
alter tablespace ts_wh_011 prefetchsize 4096;
alter tablespace ts_wh_012 prefetchsize 4096;
alter tablespace ts_wh_013 prefetchsize 4096;
alter tablespace ts_wh_014 prefetchsize 4096;
alter tablespace ts_wh_015 prefetchsize 4096;
alter tablespace ts_wh_016 prefetchsize 4096;
alter tablespace ts_wh_017 prefetchsize 4096;
alter tablespace ts_wh_018 prefetchsize 4096;
alter tablespace ts_wh_019 prefetchsize 4096;
alter tablespace ts_wh_020 prefetchsize 4096;
alter tablespace ts_wh_021 prefetchsize 4096;
alter tablespace ts_wh_022 prefetchsize 4096;
alter tablespace ts_wh_023 prefetchsize 4096;
alter tablespace ts_wh_024 prefetchsize 4096;
alter tablespace ts_wh_025 prefetchsize 4096;
alter tablespace ts_wh_026 prefetchsize 4096;
alter tablespace ts_wh_027 prefetchsize 4096;
alter tablespace ts_wh_028 prefetchsize 4096;
alter tablespace ts_wh_029 prefetchsize 4096;
alter tablespace ts_wh_030 prefetchsize 4096;
alter tablespace ts_wh_031 prefetchsize 4096;
alter tablespace ts_wh_032 prefetchsize 4096;
alter tablespace ts_wh_033 prefetchsize 4096;
alter tablespace ts_wh_034 prefetchsize 4096;
alter tablespace ts_wh_035 prefetchsize 4096;
alter tablespace ts_wh_036 prefetchsize 4096;
alter tablespace ts_wh_037 prefetchsize 4096;
alter tablespace ts_wh_038 prefetchsize 4096;
alter tablespace ts_wh_039 prefetchsize 4096;
alter tablespace ts_wh_040 prefetchsize 4096;
alter tablespace ts_wh_041 prefetchsize 4096;
alter tablespace ts_wh_042 prefetchsize 4096;
alter tablespace ts_wh_043 prefetchsize 4096;
alter tablespace ts_wh_044 prefetchsize 4096;
alter tablespace ts_wh_045 prefetchsize 4096;
alter tablespace ts_wh_046 prefetchsize 4096;
alter tablespace ts_wh_047 prefetchsize 4096;
alter tablespace ts_wh_048 prefetchsize 4096;
alter tablespace ts_wh_049 prefetchsize 4096;
alter tablespace ts_wh_050 prefetchsize 4096;
alter tablespace ts_wh_051 prefetchsize 4096;
alter tablespace ts_wh_052 prefetchsize 4096;
alter tablespace ts_wh_053 prefetchsize 4096;
alter tablespace ts_wh_054 prefetchsize 4096;
alter tablespace ts_wh_055 prefetchsize 4096;
alter tablespace ts_wh_056 prefetchsize 4096;
alter tablespace ts_wh_057 prefetchsize 4096;
alter tablespace ts_wh_058 prefetchsize 4096;
alter tablespace ts_wh_059 prefetchsize 4096;
alter tablespace ts_wh_060 prefetchsize 4096;
alter tablespace ts_wh_061 prefetchsize 4096;
alter tablespace ts_wh_062 prefetchsize 4096;
alter tablespace ts_wh_063 prefetchsize 4096;
alter tablespace ts_wh_064 prefetchsize 4096;
alter tablespace ts_wh_065 prefetchsize 4096;
alter tablespace ts_wh_066 prefetchsize 4096;
alter tablespace ts_wh_067 prefetchsize 4096;
alter tablespace ts_wh_068 prefetchsize 4096;
alter tablespace ts_wh_069 prefetchsize 4096;
alter tablespace ts_wh_070 prefetchsize 4096;
alter tablespace ts_wh_071 prefetchsize 4096;
alter tablespace ts_wh_072 prefetchsize 4096;
alter tablespace ts_wh_073 prefetchsize 4096;
alter tablespace ts_wh_074 prefetchsize 4096;
alter tablespace ts_wh_075 prefetchsize 4096;
alter tablespace ts_wh_076 prefetchsize 4096;
alter tablespace ts_wh_077 prefetchsize 4096;
alter tablespace ts_wh_078 prefetchsize 4096;
alter tablespace ts_wh_079 prefetchsize 4096;
alter tablespace ts_wh_080 prefetchsize 4096;
connect reset;
```

DDL/CRCONST_CUSTOMER.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER1 OFF;
ALTER TABLE CUSTOMER1 DROP CONSTRAINT CUSTOMER1CKC;
ALTER TABLE CUSTOMER1 ADD CONSTRAINT CUSTOMER1CKC CHECK (C_W_ID
BETWEEN 1 AND 520);
SET INTEGRITY FOR CUSTOMER1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER2 OFF;
ALTER TABLE CUSTOMER2 DROP CONSTRAINT CUSTOMER2CKC;
ALTER TABLE CUSTOMER2 ADD CONSTRAINT CUSTOMER2CKC CHECK (C_W_ID
BETWEEN 521 AND 1040);
SET INTEGRITY FOR CUSTOMER2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER3 OFF;
ALTER TABLE CUSTOMER3 DROP CONSTRAINT CUSTOMER3CKC;
ALTER TABLE CUSTOMER3 ADD CONSTRAINT CUSTOMER3CKC CHECK (C_W_ID
BETWEEN 1041 AND 1560);
SET INTEGRITY FOR CUSTOMER3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER4 OFF;
ALTER TABLE CUSTOMER4 DROP CONSTRAINT CUSTOMER4CKC;
ALTER TABLE CUSTOMER4 ADD CONSTRAINT CUSTOMER4CKC CHECK (C_W_ID
BETWEEN 1561 AND 2080);
SET INTEGRITY FOR CUSTOMER4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER5 OFF;
ALTER TABLE CUSTOMER5 DROP CONSTRAINT CUSTOMER5CKC;
ALTER TABLE CUSTOMER5 ADD CONSTRAINT CUSTOMER5CKC CHECK (C_W_ID
BETWEEN 2081 AND 2600);
SET INTEGRITY FOR CUSTOMER5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER6 OFF;
ALTER TABLE CUSTOMER6 DROP CONSTRAINT CUSTOMER6CKC;
ALTER TABLE CUSTOMER6 ADD CONSTRAINT CUSTOMER6CKC CHECK (C_W_ID
BETWEEN 2601 AND 3120);
SET INTEGRITY FOR CUSTOMER6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER7 OFF;
ALTER TABLE CUSTOMER7 DROP CONSTRAINT CUSTOMER7CKC;
ALTER TABLE CUSTOMER7 ADD CONSTRAINT CUSTOMER7CKC CHECK (C_W_ID
BETWEEN 3121 AND 3640);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER8 OFF;
ALTER TABLE CUSTOMER8 DROP CONSTRAINT CUSTOMER8CKC;
ALTER TABLE CUSTOMER8 ADD CONSTRAINT CUSTOMER8CKC CHECK (C_W_ID
BETWEEN 3641 AND 4160);
SET INTEGRITY FOR CUSTOMER8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER9 OFF;
ALTER TABLE CUSTOMER9 DROP CONSTRAINT CUSTOMER9CKC;
ALTER TABLE CUSTOMER9 ADD CONSTRAINT CUSTOMER9CKC CHECK (C_W_ID
BETWEEN 4161 AND 4680);
SET INTEGRITY FOR CUSTOMER9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER10 OFF;
ALTER TABLE CUSTOMER10 DROP CONSTRAINT CUSTOMER10CKC;
ALTER TABLE CUSTOMER10 ADD CONSTRAINT CUSTOMER10CKC CHECK (C_W_ID
BETWEEN 4681 AND 5200);
SET INTEGRITY FOR CUSTOMER10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER11 OFF;
ALTER TABLE CUSTOMER11 DROP CONSTRAINT CUSTOMER11CKC;
ALTER TABLE CUSTOMER11 ADD CONSTRAINT CUSTOMER11CKC CHECK (C_W_ID
BETWEEN 5201 AND 5720);
SET INTEGRITY FOR CUSTOMER11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER12 OFF;
ALTER TABLE CUSTOMER12 DROP CONSTRAINT CUSTOMER12CKC;
ALTER TABLE CUSTOMER12 ADD CONSTRAINT CUSTOMER12CKC CHECK (C_W_ID
BETWEEN 5721 AND 6240);
SET INTEGRITY FOR CUSTOMER12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER13 OFF;
ALTER TABLE CUSTOMER13 DROP CONSTRAINT CUSTOMER13CKC;
ALTER TABLE CUSTOMER13 ADD CONSTRAINT CUSTOMER13CKC CHECK (C_W_ID
BETWEEN 6241 AND 6760);
SET INTEGRITY FOR CUSTOMER13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER14 OFF;
ALTER TABLE CUSTOMER14 DROP CONSTRAINT CUSTOMER14CKC;
ALTER TABLE CUSTOMER14 ADD CONSTRAINT CUSTOMER14CKC CHECK (C_W_ID
BETWEEN 6761 AND 7280);
SET INTEGRITY FOR CUSTOMER14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER15 OFF;
ALTER TABLE CUSTOMER15 DROP CONSTRAINT CUSTOMER15CKC;
```

```
ALTER TABLE CUSTOMER15 ADD CONSTRAINT CUSTOMER15CKC CHECK (C_W_ID BETWEEN 7281 AND 7800);
SET INTEGRITY FOR CUSTOMER15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER16 OFF;
ALTER TABLE CUSTOMER16 DROP CONSTRAINT CUSTOMER16CKC;
ALTER TABLE CUSTOMER16 ADD CONSTRAINT CUSTOMER16CKC CHECK (C_W_ID BETWEEN 7801 AND 8320);
SET INTEGRITY FOR CUSTOMER16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER17 OFF;
ALTER TABLE CUSTOMER17 DROP CONSTRAINT CUSTOMER17CKC;
ALTER TABLE CUSTOMER17 ADD CONSTRAINT CUSTOMER17CKC CHECK (C_W_ID BETWEEN 8321 AND 8840);
SET INTEGRITY FOR CUSTOMER17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER18 OFF;
ALTER TABLE CUSTOMER18 DROP CONSTRAINT CUSTOMER18CKC;
ALTER TABLE CUSTOMER18 ADD CONSTRAINT CUSTOMER18CKC CHECK (C_W_ID BETWEEN 8841 AND 9360);
SET INTEGRITY FOR CUSTOMER18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER19 OFF;
ALTER TABLE CUSTOMER19 DROP CONSTRAINT CUSTOMER19CKC;
ALTER TABLE CUSTOMER19 ADD CONSTRAINT CUSTOMER19CKC CHECK (C_W_ID BETWEEN 9361 AND 9880);
SET INTEGRITY FOR CUSTOMER19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER20 OFF;
ALTER TABLE CUSTOMER20 DROP CONSTRAINT CUSTOMER20CKC;
ALTER TABLE CUSTOMER20 ADD CONSTRAINT CUSTOMER20CKC CHECK (C_W_ID BETWEEN 9881 AND 10400);
SET INTEGRITY FOR CUSTOMER20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER21 OFF;
ALTER TABLE CUSTOMER21 DROP CONSTRAINT CUSTOMER21CKC;
ALTER TABLE CUSTOMER21 ADD CONSTRAINT CUSTOMER21CKC CHECK (C_W_ID BETWEEN 10401 AND 10920);
SET INTEGRITY FOR CUSTOMER21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER22 OFF;
ALTER TABLE CUSTOMER22 DROP CONSTRAINT CUSTOMER22CKC;
ALTER TABLE CUSTOMER22 ADD CONSTRAINT CUSTOMER22CKC CHECK (C_W_ID BETWEEN 10921 AND 11440);
SET INTEGRITY FOR CUSTOMER22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER23 OFF;
ALTER TABLE CUSTOMER23 DROP CONSTRAINT CUSTOMER23CKC;
ALTER TABLE CUSTOMER23 ADD CONSTRAINT CUSTOMER23CKC CHECK (C_W_ID BETWEEN 11441 AND 11960);
SET INTEGRITY FOR CUSTOMER23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER24 OFF;
ALTER TABLE CUSTOMER24 DROP CONSTRAINT CUSTOMER24CKC;
ALTER TABLE CUSTOMER24 ADD CONSTRAINT CUSTOMER24CKC CHECK (C_W_ID BETWEEN 11961 AND 12480);
SET INTEGRITY FOR CUSTOMER24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER25 OFF;
ALTER TABLE CUSTOMER25 DROP CONSTRAINT CUSTOMER25CKC;
ALTER TABLE CUSTOMER25 ADD CONSTRAINT CUSTOMER25CKC CHECK (C_W_ID BETWEEN 12481 AND 13000);
SET INTEGRITY FOR CUSTOMER25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER26 OFF;
ALTER TABLE CUSTOMER26 DROP CONSTRAINT CUSTOMER26CKC;
ALTER TABLE CUSTOMER26 ADD CONSTRAINT CUSTOMER26CKC CHECK (C_W_ID BETWEEN 13001 AND 13520);
SET INTEGRITY FOR CUSTOMER26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER27 OFF;
ALTER TABLE CUSTOMER27 DROP CONSTRAINT CUSTOMER27CKC;
ALTER TABLE CUSTOMER27 ADD CONSTRAINT CUSTOMER27CKC CHECK (C_W_ID BETWEEN 13521 AND 14040);
SET INTEGRITY FOR CUSTOMER27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER28 OFF;
ALTER TABLE CUSTOMER28 DROP CONSTRAINT CUSTOMER28CKC;
ALTER TABLE CUSTOMER28 ADD CONSTRAINT CUSTOMER28CKC CHECK (C_W_ID BETWEEN 14041 AND 14560);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER29 OFF;
ALTER TABLE CUSTOMER29 DROP CONSTRAINT CUSTOMER29CKC;
ALTER TABLE CUSTOMER29 ADD CONSTRAINT CUSTOMER29CKC CHECK (C_W_ID BETWEEN 14561 AND 15080);
SET INTEGRITY FOR CUSTOMER29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER30 OFF;
ALTER TABLE CUSTOMER30 DROP CONSTRAINT CUSTOMER30CKC;
ALTER TABLE CUSTOMER30 ADD CONSTRAINT CUSTOMER30CKC CHECK (C_W_ID BETWEEN 15081 AND 15600);
SET INTEGRITY FOR CUSTOMER30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER31 OFF;
ALTER TABLE CUSTOMER31 DROP CONSTRAINT CUSTOMER31CKC;
ALTER TABLE CUSTOMER31 ADD CONSTRAINT CUSTOMER31CKC CHECK (C_W_ID BETWEEN 15601 AND 16120);
SET INTEGRITY FOR CUSTOMER31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER32 OFF;
ALTER TABLE CUSTOMER32 DROP CONSTRAINT CUSTOMER32CKC;
ALTER TABLE CUSTOMER32 ADD CONSTRAINT CUSTOMER32CKC CHECK (C_W_ID BETWEEN 16121 AND 16640);
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER33 OFF;
ALTER TABLE CUSTOMER33 DROP CONSTRAINT CUSTOMER33CKC;
ALTER TABLE CUSTOMER33 ADD CONSTRAINT CUSTOMER33CKC CHECK (C_W_ID BETWEEN 16641 AND 17160);
SET INTEGRITY FOR CUSTOMER33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER34 OFF;
ALTER TABLE CUSTOMER34 DROP CONSTRAINT CUSTOMER34CKC;
ALTER TABLE CUSTOMER34 ADD CONSTRAINT CUSTOMER34CKC CHECK (C_W_ID BETWEEN 17161 AND 17680);
SET INTEGRITY FOR CUSTOMER34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER35 OFF;
ALTER TABLE CUSTOMER35 DROP CONSTRAINT CUSTOMER35CKC;
ALTER TABLE CUSTOMER35 ADD CONSTRAINT CUSTOMER35CKC CHECK (C_W_ID BETWEEN 17681 AND 18200);
SET INTEGRITY FOR CUSTOMER35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER36 OFF;
ALTER TABLE CUSTOMER36 DROP CONSTRAINT CUSTOMER36CKC;
ALTER TABLE CUSTOMER36 ADD CONSTRAINT CUSTOMER36CKC CHECK (C_W_ID BETWEEN 18201 AND 18720);
SET INTEGRITY FOR CUSTOMER36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER37 OFF;
ALTER TABLE CUSTOMER37 DROP CONSTRAINT CUSTOMER37CKC;
ALTER TABLE CUSTOMER37 ADD CONSTRAINT CUSTOMER37CKC CHECK (C_W_ID BETWEEN 18721 AND 19240);
SET INTEGRITY FOR CUSTOMER37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER38 OFF;
ALTER TABLE CUSTOMER38 DROP CONSTRAINT CUSTOMER38CKC;
ALTER TABLE CUSTOMER38 ADD CONSTRAINT CUSTOMER38CKC CHECK (C_W_ID BETWEEN 19241 AND 19760);
SET INTEGRITY FOR CUSTOMER38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER39 OFF;
ALTER TABLE CUSTOMER39 DROP CONSTRAINT CUSTOMER39CKC;
ALTER TABLE CUSTOMER39 ADD CONSTRAINT CUSTOMER39CKC CHECK (C_W_ID BETWEEN 19761 AND 20280);
SET INTEGRITY FOR CUSTOMER39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER40 OFF;
ALTER TABLE CUSTOMER40 DROP CONSTRAINT CUSTOMER40CKC;
ALTER TABLE CUSTOMER40 ADD CONSTRAINT CUSTOMER40CKC CHECK (C_W_ID BETWEEN 20281 AND 20800);
SET INTEGRITY FOR CUSTOMER40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER41 OFF;
ALTER TABLE CUSTOMER41 DROP CONSTRAINT CUSTOMER41CKC;
ALTER TABLE CUSTOMER41 ADD CONSTRAINT CUSTOMER41CKC CHECK (C_W_ID BETWEEN 20801 AND 21320);
SET INTEGRITY FOR CUSTOMER41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER42 OFF;
ALTER TABLE CUSTOMER42 DROP CONSTRAINT CUSTOMER42CKC;
ALTER TABLE CUSTOMER42 ADD CONSTRAINT CUSTOMER42CKC CHECK (C_W_ID BETWEEN 21321 AND 21840);
SET INTEGRITY FOR CUSTOMER42 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER43 OFF;
ALTER TABLE CUSTOMER43 DROP CONSTRAINT CUSTOMER43CKC;
ALTER TABLE CUSTOMER43 ADD CONSTRAINT CUSTOMER43CKC CHECK (C_W_ID BETWEEN 21841 AND 22360);
SET INTEGRITY FOR CUSTOMER43 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER44 OFF;
ALTER TABLE CUSTOMER44 DROP CONSTRAINT CUSTOMER44CKC;
ALTER TABLE CUSTOMER44 ADD CONSTRAINT CUSTOMER44CKC CHECK (C_W_ID BETWEEN 22361 AND 22880);
SET INTEGRITY FOR CUSTOMER44 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER45 OFF;
ALTER TABLE CUSTOMER45 DROP CONSTRAINT CUSTOMER45CKC;
```

```
ALTER TABLE CUSTOMER45 ADD CONSTRAINT CUSTOMER45CKC CHECK (C_W_ID BETWEEN 22881 AND 23400);
SET INTEGRITY FOR CUSTOMER45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER46 OFF;
ALTER TABLE CUSTOMER46 DROP CONSTRAINT CUSTOMER46CKC;
ALTER TABLE CUSTOMER46 ADD CONSTRAINT CUSTOMER46CKC CHECK (C_W_ID BETWEEN 23401 AND 23920);
SET INTEGRITY FOR CUSTOMER46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER47 OFF;
ALTER TABLE CUSTOMER47 DROP CONSTRAINT CUSTOMER47CKC;
ALTER TABLE CUSTOMER47 ADD CONSTRAINT CUSTOMER47CKC CHECK (C_W_ID BETWEEN 23921 AND 24440);
SET INTEGRITY FOR CUSTOMER47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER48 OFF;
ALTER TABLE CUSTOMER48 DROP CONSTRAINT CUSTOMER48CKC;
ALTER TABLE CUSTOMER48 ADD CONSTRAINT CUSTOMER48CKC CHECK (C_W_ID BETWEEN 24441 AND 24960);
SET INTEGRITY FOR CUSTOMER48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER49 OFF;
ALTER TABLE CUSTOMER49 DROP CONSTRAINT CUSTOMER49CKC;
ALTER TABLE CUSTOMER49 ADD CONSTRAINT CUSTOMER49CKC CHECK (C_W_ID BETWEEN 24961 AND 25480);
SET INTEGRITY FOR CUSTOMER49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER50 OFF;
ALTER TABLE CUSTOMER50 DROP CONSTRAINT CUSTOMER50CKC;
ALTER TABLE CUSTOMER50 ADD CONSTRAINT CUSTOMER50CKC CHECK (C_W_ID BETWEEN 25481 AND 26000);
SET INTEGRITY FOR CUSTOMER50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER51 OFF;
ALTER TABLE CUSTOMER51 DROP CONSTRAINT CUSTOMER51CKC;
ALTER TABLE CUSTOMER51 ADD CONSTRAINT CUSTOMER51CKC CHECK (C_W_ID BETWEEN 26001 AND 26520);
SET INTEGRITY FOR CUSTOMER51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER52 OFF;
ALTER TABLE CUSTOMER52 DROP CONSTRAINT CUSTOMER52CKC;
ALTER TABLE CUSTOMER52 ADD CONSTRAINT CUSTOMER52CKC CHECK (C_W_ID BETWEEN 26521 AND 27040);
SET INTEGRITY FOR CUSTOMER52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER53 OFF;
ALTER TABLE CUSTOMER53 DROP CONSTRAINT CUSTOMER53CKC;
ALTER TABLE CUSTOMER53 ADD CONSTRAINT CUSTOMER53CKC CHECK (C_W_ID BETWEEN 27041 AND 27560);
SET INTEGRITY FOR CUSTOMER53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER54 OFF;
ALTER TABLE CUSTOMER54 DROP CONSTRAINT CUSTOMER54CKC;
ALTER TABLE CUSTOMER54 ADD CONSTRAINT CUSTOMER54CKC CHECK (C_W_ID BETWEEN 27561 AND 28080);
SET INTEGRITY FOR CUSTOMER54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER55 OFF;
ALTER TABLE CUSTOMER55 DROP CONSTRAINT CUSTOMER55CKC;
ALTER TABLE CUSTOMER55 ADD CONSTRAINT CUSTOMER55CKC CHECK (C_W_ID BETWEEN 28081 AND 28600);
SET INTEGRITY FOR CUSTOMER55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER56 OFF;
ALTER TABLE CUSTOMER56 DROP CONSTRAINT CUSTOMER56CKC;
ALTER TABLE CUSTOMER56 ADD CONSTRAINT CUSTOMER56CKC CHECK (C_W_ID BETWEEN 28601 AND 29120);
SET INTEGRITY FOR CUSTOMER56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER57 OFF;
ALTER TABLE CUSTOMER57 DROP CONSTRAINT CUSTOMER57CKC;
ALTER TABLE CUSTOMER57 ADD CONSTRAINT CUSTOMER57CKC CHECK (C_W_ID BETWEEN 29121 AND 29640);
SET INTEGRITY FOR CUSTOMER57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER58 OFF;
ALTER TABLE CUSTOMER58 DROP CONSTRAINT CUSTOMER58CKC;
ALTER TABLE CUSTOMER58 ADD CONSTRAINT CUSTOMER58CKC CHECK (C_W_ID BETWEEN 29641 AND 30160);
SET INTEGRITY FOR CUSTOMER58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER59 OFF;
ALTER TABLE CUSTOMER59 DROP CONSTRAINT CUSTOMER59CKC;
ALTER TABLE CUSTOMER59 ADD CONSTRAINT CUSTOMER59CKC CHECK (C_W_ID BETWEEN 30161 AND 30680);
SET INTEGRITY FOR CUSTOMER59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER60 OFF;
ALTER TABLE CUSTOMER60 DROP CONSTRAINT CUSTOMER60CKC;
ALTER TABLE CUSTOMER60 ADD CONSTRAINT CUSTOMER60CKC CHECK (C_W_ID BETWEEN 30681 AND 31200);
SET INTEGRITY FOR CUSTOMER60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER61 OFF;
ALTER TABLE CUSTOMER61 DROP CONSTRAINT CUSTOMER61CKC;
ALTER TABLE CUSTOMER61 ADD CONSTRAINT CUSTOMER61CKC CHECK (C_W_ID BETWEEN 31201 AND 31720);
SET INTEGRITY FOR CUSTOMER61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER62 OFF;
ALTER TABLE CUSTOMER62 DROP CONSTRAINT CUSTOMER62CKC;
ALTER TABLE CUSTOMER62 ADD CONSTRAINT CUSTOMER62CKC CHECK (C_W_ID BETWEEN 31721 AND 32240);
SET INTEGRITY FOR CUSTOMER62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER63 OFF;
ALTER TABLE CUSTOMER63 DROP CONSTRAINT CUSTOMER63CKC;
ALTER TABLE CUSTOMER63 ADD CONSTRAINT CUSTOMER63CKC CHECK (C_W_ID BETWEEN 32241 AND 32760);
SET INTEGRITY FOR CUSTOMER63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER64 OFF;
ALTER TABLE CUSTOMER64 DROP CONSTRAINT CUSTOMER64CKC;
ALTER TABLE CUSTOMER64 ADD CONSTRAINT CUSTOMER64CKC CHECK (C_W_ID BETWEEN 32761 AND 33280);
SET INTEGRITY FOR CUSTOMER64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER65 OFF;
ALTER TABLE CUSTOMER65 DROP CONSTRAINT CUSTOMER65CKC;
ALTER TABLE CUSTOMER65 ADD CONSTRAINT CUSTOMER65CKC CHECK (C_W_ID BETWEEN 33281 AND 33800);
SET INTEGRITY FOR CUSTOMER65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER66 OFF;
ALTER TABLE CUSTOMER66 DROP CONSTRAINT CUSTOMER66CKC;
ALTER TABLE CUSTOMER66 ADD CONSTRAINT CUSTOMER66CKC CHECK (C_W_ID BETWEEN 33801 AND 34320);
SET INTEGRITY FOR CUSTOMER66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER67 OFF;
ALTER TABLE CUSTOMER67 DROP CONSTRAINT CUSTOMER67CKC;
ALTER TABLE CUSTOMER67 ADD CONSTRAINT CUSTOMER67CKC CHECK (C_W_ID BETWEEN 34321 AND 34840);
SET INTEGRITY FOR CUSTOMER67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER68 OFF;
ALTER TABLE CUSTOMER68 DROP CONSTRAINT CUSTOMER68CKC;
ALTER TABLE CUSTOMER68 ADD CONSTRAINT CUSTOMER68CKC CHECK (C_W_ID BETWEEN 34841 AND 35360);
SET INTEGRITY FOR CUSTOMER68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER69 OFF;
ALTER TABLE CUSTOMER69 DROP CONSTRAINT CUSTOMER69CKC;
ALTER TABLE CUSTOMER69 ADD CONSTRAINT CUSTOMER69CKC CHECK (C_W_ID BETWEEN 35361 AND 35880);
SET INTEGRITY FOR CUSTOMER69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER70 OFF;
ALTER TABLE CUSTOMER70 DROP CONSTRAINT CUSTOMER70CKC;
ALTER TABLE CUSTOMER70 ADD CONSTRAINT CUSTOMER70CKC CHECK (C_W_ID BETWEEN 35881 AND 36400);
SET INTEGRITY FOR CUSTOMER70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER71 OFF;
ALTER TABLE CUSTOMER71 DROP CONSTRAINT CUSTOMER71CKC;
ALTER TABLE CUSTOMER71 ADD CONSTRAINT CUSTOMER71CKC CHECK (C_W_ID BETWEEN 36401 AND 36920);
SET INTEGRITY FOR CUSTOMER71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER72 OFF;
ALTER TABLE CUSTOMER72 DROP CONSTRAINT CUSTOMER72CKC;
ALTER TABLE CUSTOMER72 ADD CONSTRAINT CUSTOMER72CKC CHECK (C_W_ID BETWEEN 36921 AND 37440);
SET INTEGRITY FOR CUSTOMER72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER73 OFF;
ALTER TABLE CUSTOMER73 DROP CONSTRAINT CUSTOMER73CKC;
ALTER TABLE CUSTOMER73 ADD CONSTRAINT CUSTOMER73CKC CHECK (C_W_ID BETWEEN 37441 AND 37960);
SET INTEGRITY FOR CUSTOMER73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER74 OFF;
ALTER TABLE CUSTOMER74 DROP CONSTRAINT CUSTOMER74CKC;
ALTER TABLE CUSTOMER74 ADD CONSTRAINT CUSTOMER74CKC CHECK (C_W_ID BETWEEN 37961 AND 38480);
SET INTEGRITY FOR CUSTOMER74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER75 OFF;
ALTER TABLE CUSTOMER75 DROP CONSTRAINT CUSTOMER75CKC;
```

```
ALTER TABLE CUSTOMER75 ADD CONSTRAINT CUSTOMER75CKC CHECK (C_W_ID
BETWEEN 38481 AND 39000);
SET INTEGRITY FOR CUSTOMER75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER76 OFF;
ALTER TABLE CUSTOMER76 DROP CONSTRAINT CUSTOMER76CKC;
ALTER TABLE CUSTOMER76 ADD CONSTRAINT CUSTOMER76CKC CHECK (C_W_ID
BETWEEN 39001 AND 39520);
SET INTEGRITY FOR CUSTOMER76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER77 OFF;
ALTER TABLE CUSTOMER77 DROP CONSTRAINT CUSTOMER77CKC;
ALTER TABLE CUSTOMER77 ADD CONSTRAINT CUSTOMER77CKC CHECK (C_W_ID
BETWEEN 39521 AND 40040);
SET INTEGRITY FOR CUSTOMER77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER78 OFF;
ALTER TABLE CUSTOMER78 DROP CONSTRAINT CUSTOMER78CKC;
ALTER TABLE CUSTOMER78 ADD CONSTRAINT CUSTOMER78CKC CHECK (C_W_ID
BETWEEN 40041 AND 40560);
SET INTEGRITY FOR CUSTOMER78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER79 OFF;
ALTER TABLE CUSTOMER79 DROP CONSTRAINT CUSTOMER79CKC;
ALTER TABLE CUSTOMER79 ADD CONSTRAINT CUSTOMER79CKC CHECK (C_W_ID
BETWEEN 40561 AND 41080);
SET INTEGRITY FOR CUSTOMER79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER80 OFF;
ALTER TABLE CUSTOMER80 DROP CONSTRAINT CUSTOMER80CKC;
ALTER TABLE CUSTOMER80 ADD CONSTRAINT CUSTOMER80CKC CHECK (C_W_ID
>= 41081);
SET INTEGRITY FOR CUSTOMER80 ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_DISTRICT.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT1 OFF;
ALTER TABLE DISTRICT1 DROP CONSTRAINT DISTRICT1CKC;
ALTER TABLE DISTRICT1 ADD CONSTRAINT DISTRICT1CKC CHECK (D_W_ID
BETWEEN 1 AND 520);
SET INTEGRITY FOR DISTRICT1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT2 OFF;
ALTER TABLE DISTRICT2 DROP CONSTRAINT DISTRICT2CKC;
ALTER TABLE DISTRICT2 ADD CONSTRAINT DISTRICT2CKC CHECK (D_W_ID
BETWEEN 521 AND 1040);
SET INTEGRITY FOR DISTRICT2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT3 OFF;
ALTER TABLE DISTRICT3 DROP CONSTRAINT DISTRICT3CKC;
ALTER TABLE DISTRICT3 ADD CONSTRAINT DISTRICT3CKC CHECK (D_W_ID
BETWEEN 1041 AND 1560);
SET INTEGRITY FOR DISTRICT3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT4 OFF;
ALTER TABLE DISTRICT4 DROP CONSTRAINT DISTRICT4CKC;
ALTER TABLE DISTRICT4 ADD CONSTRAINT DISTRICT4CKC CHECK (D_W_ID
BETWEEN 1561 AND 2080);
SET INTEGRITY FOR DISTRICT4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT5 OFF;
ALTER TABLE DISTRICT5 DROP CONSTRAINT DISTRICT5CKC;
ALTER TABLE DISTRICT5 ADD CONSTRAINT DISTRICT5CKC CHECK (D_W_ID
BETWEEN 2081 AND 2600);
SET INTEGRITY FOR DISTRICT5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT6 OFF;
ALTER TABLE DISTRICT6 DROP CONSTRAINT DISTRICT6CKC;
ALTER TABLE DISTRICT6 ADD CONSTRAINT DISTRICT6CKC CHECK (D_W_ID
BETWEEN 2601 AND 3120);
SET INTEGRITY FOR DISTRICT6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT7 OFF;
ALTER TABLE DISTRICT7 DROP CONSTRAINT DISTRICT7CKC;
ALTER TABLE DISTRICT7 ADD CONSTRAINT DISTRICT7CKC CHECK (D_W_ID
BETWEEN 3121 AND 3640);
SET INTEGRITY FOR DISTRICT7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT8 OFF;
ALTER TABLE DISTRICT8 DROP CONSTRAINT DISTRICT8CKC;
ALTER TABLE DISTRICT8 ADD CONSTRAINT DISTRICT8CKC CHECK (D_W_ID
BETWEEN 3641 AND 4160);
SET INTEGRITY FOR DISTRICT8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT9 OFF;
ALTER TABLE DISTRICT9 DROP CONSTRAINT DISTRICT9CKC;
ALTER TABLE DISTRICT9 ADD CONSTRAINT DISTRICT9CKC CHECK (D_W_ID
BETWEEN 4161 AND 4680);
SET INTEGRITY FOR DISTRICT9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT10 OFF;
ALTER TABLE DISTRICT10 DROP CONSTRAINT DISTRICT10CKC;
ALTER TABLE DISTRICT10 ADD CONSTRAINT DISTRICT10CKC CHECK (D_W_ID
BETWEEN 4681 AND 5200);
SET INTEGRITY FOR DISTRICT10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT11 OFF;
ALTER TABLE DISTRICT11 DROP CONSTRAINT DISTRICT11CKC;
ALTER TABLE DISTRICT11 ADD CONSTRAINT DISTRICT11CKC CHECK (D_W_ID
BETWEEN 5201 AND 5720);
SET INTEGRITY FOR DISTRICT11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT12 OFF;
ALTER TABLE DISTRICT12 DROP CONSTRAINT DISTRICT12CKC;
ALTER TABLE DISTRICT12 ADD CONSTRAINT DISTRICT12CKC CHECK (D_W_ID
BETWEEN 5721 AND 6240);
SET INTEGRITY FOR DISTRICT12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT13 OFF;
ALTER TABLE DISTRICT13 DROP CONSTRAINT DISTRICT13CKC;
ALTER TABLE DISTRICT13 ADD CONSTRAINT DISTRICT13CKC CHECK (D_W_ID
BETWEEN 6241 AND 6760);
SET INTEGRITY FOR DISTRICT13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT14 OFF;
ALTER TABLE DISTRICT14 DROP CONSTRAINT DISTRICT14CKC;
ALTER TABLE DISTRICT14 ADD CONSTRAINT DISTRICT14CKC CHECK (D_W_ID
BETWEEN 6761 AND 7280);
SET INTEGRITY FOR DISTRICT14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT15 OFF;
ALTER TABLE DISTRICT15 DROP CONSTRAINT DISTRICT15CKC;
ALTER TABLE DISTRICT15 ADD CONSTRAINT DISTRICT15CKC CHECK (D_W_ID
BETWEEN 7281 AND 7800);
SET INTEGRITY FOR DISTRICT15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT16 OFF;
ALTER TABLE DISTRICT16 DROP CONSTRAINT DISTRICT16CKC;
ALTER TABLE DISTRICT16 ADD CONSTRAINT DISTRICT16CKC CHECK (D_W_ID
BETWEEN 7801 AND 8320);
SET INTEGRITY FOR DISTRICT16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT17 OFF;
ALTER TABLE DISTRICT17 DROP CONSTRAINT DISTRICT17CKC;
ALTER TABLE DISTRICT17 ADD CONSTRAINT DISTRICT17CKC CHECK (D_W_ID
BETWEEN 8321 AND 8840);
SET INTEGRITY FOR DISTRICT17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT18 OFF;
ALTER TABLE DISTRICT18 DROP CONSTRAINT DISTRICT18CKC;
ALTER TABLE DISTRICT18 ADD CONSTRAINT DISTRICT18CKC CHECK (D_W_ID
BETWEEN 8841 AND 9360);
SET INTEGRITY FOR DISTRICT18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT19 OFF;
ALTER TABLE DISTRICT19 DROP CONSTRAINT DISTRICT19CKC;
ALTER TABLE DISTRICT19 ADD CONSTRAINT DISTRICT19CKC CHECK (D_W_ID
BETWEEN 9361 AND 9880);
SET INTEGRITY FOR DISTRICT19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT20 OFF;
ALTER TABLE DISTRICT20 DROP CONSTRAINT DISTRICT20CKC;
ALTER TABLE DISTRICT20 ADD CONSTRAINT DISTRICT20CKC CHECK (D_W_ID
BETWEEN 9881 AND 10400);
SET INTEGRITY FOR DISTRICT20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT21 OFF;
ALTER TABLE DISTRICT21 DROP CONSTRAINT DISTRICT21CKC;
ALTER TABLE DISTRICT21 ADD CONSTRAINT DISTRICT21CKC CHECK (D_W_ID
BETWEEN 10401 AND 10920);
SET INTEGRITY FOR DISTRICT21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT22 OFF;
ALTER TABLE DISTRICT22 DROP CONSTRAINT DISTRICT22CKC;
ALTER TABLE DISTRICT22 ADD CONSTRAINT DISTRICT22CKC CHECK (D_W_ID
BETWEEN 10921 AND 11440);
SET INTEGRITY FOR DISTRICT22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT23 OFF;
ALTER TABLE DISTRICT23 DROP CONSTRAINT DISTRICT23CKC;
ALTER TABLE DISTRICT23 ADD CONSTRAINT DISTRICT23CKC CHECK (D_W_ID
BETWEEN 11441 AND 11960);
SET INTEGRITY FOR DISTRICT23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT24 OFF;
ALTER TABLE DISTRICT24 DROP CONSTRAINT DISTRICT24CKC;
ALTER TABLE DISTRICT24 ADD CONSTRAINT DISTRICT24CKC CHECK (D_W_ID
BETWEEN 11961 AND 12480);
SET INTEGRITY FOR DISTRICT24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT25 OFF;
ALTER TABLE DISTRICT25 DROP CONSTRAINT DISTRICT25CKC;
ALTER TABLE DISTRICT25 ADD CONSTRAINT DISTRICT25CKC CHECK (D_W_ID
BETWEEN 12481 AND 13000);
SET INTEGRITY FOR DISTRICT25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT26 OFF;
ALTER TABLE DISTRICT26 DROP CONSTRAINT DISTRICT26CKC;
ALTER TABLE DISTRICT26 ADD CONSTRAINT DISTRICT26CKC CHECK (D_W_ID
BETWEEN 13001 AND 13520);
SET INTEGRITY FOR DISTRICT26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT27 OFF;
ALTER TABLE DISTRICT27 DROP CONSTRAINT DISTRICT27CKC;
ALTER TABLE DISTRICT27 ADD CONSTRAINT DISTRICT27CKC CHECK (D_W_ID
BETWEEN 13521 AND 14040);
SET INTEGRITY FOR DISTRICT27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT28 OFF;
ALTER TABLE DISTRICT28 DROP CONSTRAINT DISTRICT28CKC;
ALTER TABLE DISTRICT28 ADD CONSTRAINT DISTRICT28CKC CHECK (D_W_ID
BETWEEN 14041 AND 14560);
SET INTEGRITY FOR DISTRICT28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT29 OFF;
ALTER TABLE DISTRICT29 DROP CONSTRAINT DISTRICT29CKC;
ALTER TABLE DISTRICT29 ADD CONSTRAINT DISTRICT29CKC CHECK (D_W_ID
BETWEEN 14561 AND 15080);
SET INTEGRITY FOR DISTRICT29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT30 OFF;
ALTER TABLE DISTRICT30 DROP CONSTRAINT DISTRICT30CKC;
ALTER TABLE DISTRICT30 ADD CONSTRAINT DISTRICT30CKC CHECK (D_W_ID
BETWEEN 15081 AND 15600);
SET INTEGRITY FOR DISTRICT30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT31 OFF;
ALTER TABLE DISTRICT31 DROP CONSTRAINT DISTRICT31CKC;
ALTER TABLE DISTRICT31 ADD CONSTRAINT DISTRICT31CKC CHECK (D_W_ID
BETWEEN 15601 AND 16120);
SET INTEGRITY FOR DISTRICT31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT32 OFF;
ALTER TABLE DISTRICT32 DROP CONSTRAINT DISTRICT32CKC;
ALTER TABLE DISTRICT32 ADD CONSTRAINT DISTRICT32CKC CHECK (D_W_ID
BETWEEN 16121 AND 16640);
SET INTEGRITY FOR DISTRICT32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT33 OFF;
ALTER TABLE DISTRICT33 DROP CONSTRAINT DISTRICT33CKC;
ALTER TABLE DISTRICT33 ADD CONSTRAINT DISTRICT33CKC CHECK (D_W_ID
BETWEEN 16641 AND 17160);
SET INTEGRITY FOR DISTRICT33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT34 OFF;
ALTER TABLE DISTRICT34 DROP CONSTRAINT DISTRICT34CKC;
ALTER TABLE DISTRICT34 ADD CONSTRAINT DISTRICT34CKC CHECK (D_W_ID
BETWEEN 17161 AND 17680);
SET INTEGRITY FOR DISTRICT34 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT35 OFF;
ALTER TABLE DISTRICT35 DROP CONSTRAINT DISTRICT35CKC;
ALTER TABLE DISTRICT35 ADD CONSTRAINT DISTRICT35CKC CHECK (D_W_ID
BETWEEN 17681 AND 18200);
SET INTEGRITY FOR DISTRICT35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT36 OFF;
ALTER TABLE DISTRICT36 DROP CONSTRAINT DISTRICT36CKC;
ALTER TABLE DISTRICT36 ADD CONSTRAINT DISTRICT36CKC CHECK (D_W_ID
BETWEEN 18201 AND 18720);
SET INTEGRITY FOR DISTRICT36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT37 OFF;
ALTER TABLE DISTRICT37 DROP CONSTRAINT DISTRICT37CKC;
ALTER TABLE DISTRICT37 ADD CONSTRAINT DISTRICT37CKC CHECK (D_W_ID
BETWEEN 18721 AND 19240);
SET INTEGRITY FOR DISTRICT37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT38 OFF;
ALTER TABLE DISTRICT38 DROP CONSTRAINT DISTRICT38CKC;
ALTER TABLE DISTRICT38 ADD CONSTRAINT DISTRICT38CKC CHECK (D_W_ID
BETWEEN 19241 AND 19760);
SET INTEGRITY FOR DISTRICT38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT39 OFF;
ALTER TABLE DISTRICT39 DROP CONSTRAINT DISTRICT39CKC;
ALTER TABLE DISTRICT39 ADD CONSTRAINT DISTRICT39CKC CHECK (D_W_ID
BETWEEN 19761 AND 20280);
SET INTEGRITY FOR DISTRICT39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT40 OFF;
ALTER TABLE DISTRICT40 DROP CONSTRAINT DISTRICT40CKC;
ALTER TABLE DISTRICT40 ADD CONSTRAINT DISTRICT40CKC CHECK (D_W_ID
BETWEEN 20281 AND 20800);
SET INTEGRITY FOR DISTRICT40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT41 OFF;
ALTER TABLE DISTRICT41 DROP CONSTRAINT DISTRICT41CKC;
ALTER TABLE DISTRICT41 ADD CONSTRAINT DISTRICT41CKC CHECK (D_W_ID
BETWEEN 20801 AND 21320);
SET INTEGRITY FOR DISTRICT41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT42 OFF;
ALTER TABLE DISTRICT42 DROP CONSTRAINT DISTRICT42CKC;
ALTER TABLE DISTRICT42 ADD CONSTRAINT DISTRICT42CKC CHECK (D_W_ID
BETWEEN 21321 AND 21840);
SET INTEGRITY FOR DISTRICT42 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT43 OFF;
ALTER TABLE DISTRICT43 DROP CONSTRAINT DISTRICT43CKC;
ALTER TABLE DISTRICT43 ADD CONSTRAINT DISTRICT43CKC CHECK (D_W_ID
BETWEEN 21841 AND 22360);
SET INTEGRITY FOR DISTRICT43 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT44 OFF;
ALTER TABLE DISTRICT44 DROP CONSTRAINT DISTRICT44CKC;
ALTER TABLE DISTRICT44 ADD CONSTRAINT DISTRICT44CKC CHECK (D_W_ID
BETWEEN 22361 AND 22880);
SET INTEGRITY FOR DISTRICT44 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT45 OFF;
ALTER TABLE DISTRICT45 DROP CONSTRAINT DISTRICT45CKC;
ALTER TABLE DISTRICT45 ADD CONSTRAINT DISTRICT45CKC CHECK (D_W_ID
BETWEEN 22881 AND 23400);
SET INTEGRITY FOR DISTRICT45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT46 OFF;
ALTER TABLE DISTRICT46 DROP CONSTRAINT DISTRICT46CKC;
ALTER TABLE DISTRICT46 ADD CONSTRAINT DISTRICT46CKC CHECK (D_W_ID
BETWEEN 23401 AND 23920);
SET INTEGRITY FOR DISTRICT46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT47 OFF;
ALTER TABLE DISTRICT47 DROP CONSTRAINT DISTRICT47CKC;
ALTER TABLE DISTRICT47 ADD CONSTRAINT DISTRICT47CKC CHECK (D_W_ID
BETWEEN 23921 AND 24440);
SET INTEGRITY FOR DISTRICT47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT48 OFF;
ALTER TABLE DISTRICT48 DROP CONSTRAINT DISTRICT48CKC;
ALTER TABLE DISTRICT48 ADD CONSTRAINT DISTRICT48CKC CHECK (D_W_ID
BETWEEN 24441 AND 24960);
SET INTEGRITY FOR DISTRICT48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT49 OFF;
ALTER TABLE DISTRICT49 DROP CONSTRAINT DISTRICT49CKC;
ALTER TABLE DISTRICT49 ADD CONSTRAINT DISTRICT49CKC CHECK (D_W_ID
BETWEEN 24961 AND 25480);
SET INTEGRITY FOR DISTRICT49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT50 OFF;
ALTER TABLE DISTRICT50 DROP CONSTRAINT DISTRICT50CKC;
ALTER TABLE DISTRICT50 ADD CONSTRAINT DISTRICT50CKC CHECK (D_W_ID
BETWEEN 25481 AND 26000);
SET INTEGRITY FOR DISTRICT50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT51 OFF;
ALTER TABLE DISTRICT51 DROP CONSTRAINT DISTRICT51CKC;
ALTER TABLE DISTRICT51 ADD CONSTRAINT DISTRICT51CKC CHECK (D_W_ID
BETWEEN 26001 AND 26520);
SET INTEGRITY FOR DISTRICT51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT52 OFF;
ALTER TABLE DISTRICT52 DROP CONSTRAINT DISTRICT52CKC;
ALTER TABLE DISTRICT52 ADD CONSTRAINT DISTRICT52CKC CHECK (D_W_ID
BETWEEN 26521 AND 27040);
SET INTEGRITY FOR DISTRICT52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT53 OFF;
ALTER TABLE DISTRICT53 DROP CONSTRAINT DISTRICT53CKC;
ALTER TABLE DISTRICT53 ADD CONSTRAINT DISTRICT53CKC CHECK (D_W_ID
BETWEEN 27041 AND 27560);
SET INTEGRITY FOR DISTRICT53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT54 OFF;
ALTER TABLE DISTRICT54 DROP CONSTRAINT DISTRICT54CKC;
ALTER TABLE DISTRICT54 ADD CONSTRAINT DISTRICT54CKC CHECK (D_W_ID
BETWEEN 27561 AND 28080);
SET INTEGRITY FOR DISTRICT54 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT55 OFF;
ALTER TABLE DISTRICT55 DROP CONSTRAINT DISTRICT55CKC;
ALTER TABLE DISTRICT55 ADD CONSTRAINT DISTRICT55CKC CHECK (D_W_ID
BETWEEN 28081 AND 28600);
SET INTEGRITY FOR DISTRICT55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT56 OFF;
ALTER TABLE DISTRICT56 DROP CONSTRAINT DISTRICT56CKC;
ALTER TABLE DISTRICT56 ADD CONSTRAINT DISTRICT56CKC CHECK (D_W_ID
BETWEEN 28601 AND 29120);
SET INTEGRITY FOR DISTRICT56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT57 OFF;
ALTER TABLE DISTRICT57 DROP CONSTRAINT DISTRICT57CKC;
ALTER TABLE DISTRICT57 ADD CONSTRAINT DISTRICT57CKC CHECK (D_W_ID
BETWEEN 29121 AND 29640);
SET INTEGRITY FOR DISTRICT57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT58 OFF;
ALTER TABLE DISTRICT58 DROP CONSTRAINT DISTRICT58CKC;
ALTER TABLE DISTRICT58 ADD CONSTRAINT DISTRICT58CKC CHECK (D_W_ID
BETWEEN 29641 AND 30160);
SET INTEGRITY FOR DISTRICT58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT59 OFF;
ALTER TABLE DISTRICT59 DROP CONSTRAINT DISTRICT59CKC;
ALTER TABLE DISTRICT59 ADD CONSTRAINT DISTRICT59CKC CHECK (D_W_ID
BETWEEN 30161 AND 30680);
SET INTEGRITY FOR DISTRICT59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT60 OFF;
ALTER TABLE DISTRICT60 DROP CONSTRAINT DISTRICT60CKC;
ALTER TABLE DISTRICT60 ADD CONSTRAINT DISTRICT60CKC CHECK (D_W_ID
BETWEEN 30681 AND 31200);
SET INTEGRITY FOR DISTRICT60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT61 OFF;
ALTER TABLE DISTRICT61 DROP CONSTRAINT DISTRICT61CKC;
ALTER TABLE DISTRICT61 ADD CONSTRAINT DISTRICT61CKC CHECK (D_W_ID
BETWEEN 31201 AND 31720);
SET INTEGRITY FOR DISTRICT61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT62 OFF;
ALTER TABLE DISTRICT62 DROP CONSTRAINT DISTRICT62CKC;
ALTER TABLE DISTRICT62 ADD CONSTRAINT DISTRICT62CKC CHECK (D_W_ID
BETWEEN 31721 AND 32240);
SET INTEGRITY FOR DISTRICT62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT63 OFF;
ALTER TABLE DISTRICT63 DROP CONSTRAINT DISTRICT63CKC;
ALTER TABLE DISTRICT63 ADD CONSTRAINT DISTRICT63CKC CHECK (D_W_ID
BETWEEN 32241 AND 32760);
SET INTEGRITY FOR DISTRICT63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT64 OFF;
ALTER TABLE DISTRICT64 DROP CONSTRAINT DISTRICT64CKC;
ALTER TABLE DISTRICT64 ADD CONSTRAINT DISTRICT64CKC CHECK (D_W_ID
BETWEEN 32761 AND 33280);
SET INTEGRITY FOR DISTRICT64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT65 OFF;
ALTER TABLE DISTRICT65 DROP CONSTRAINT DISTRICT65CKC;
ALTER TABLE DISTRICT65 ADD CONSTRAINT DISTRICT65CKC CHECK (D_W_ID
BETWEEN 33281 AND 33800);
SET INTEGRITY FOR DISTRICT65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT66 OFF;
ALTER TABLE DISTRICT66 DROP CONSTRAINT DISTRICT66CKC;
ALTER TABLE DISTRICT66 ADD CONSTRAINT DISTRICT66CKC CHECK (D_W_ID
BETWEEN 33801 AND 34320);
SET INTEGRITY FOR DISTRICT66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT67 OFF;
ALTER TABLE DISTRICT67 DROP CONSTRAINT DISTRICT67CKC;
ALTER TABLE DISTRICT67 ADD CONSTRAINT DISTRICT67CKC CHECK (D_W_ID
BETWEEN 34321 AND 34840);
SET INTEGRITY FOR DISTRICT67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT68 OFF;
ALTER TABLE DISTRICT68 DROP CONSTRAINT DISTRICT68CKC;
ALTER TABLE DISTRICT68 ADD CONSTRAINT DISTRICT68CKC CHECK (D_W_ID
BETWEEN 34841 AND 35360);
SET INTEGRITY FOR DISTRICT68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT69 OFF;
ALTER TABLE DISTRICT69 DROP CONSTRAINT DISTRICT69CKC;
ALTER TABLE DISTRICT69 ADD CONSTRAINT DISTRICT69CKC CHECK (D_W_ID
BETWEEN 35361 AND 35880);
SET INTEGRITY FOR DISTRICT69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT70 OFF;
ALTER TABLE DISTRICT70 DROP CONSTRAINT DISTRICT70CKC;
ALTER TABLE DISTRICT70 ADD CONSTRAINT DISTRICT70CKC CHECK (D_W_ID
BETWEEN 35881 AND 36400);
SET INTEGRITY FOR DISTRICT70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT71 OFF;
ALTER TABLE DISTRICT71 DROP CONSTRAINT DISTRICT71CKC;
ALTER TABLE DISTRICT71 ADD CONSTRAINT DISTRICT71CKC CHECK (D_W_ID
BETWEEN 36401 AND 36920);
SET INTEGRITY FOR DISTRICT71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT72 OFF;
ALTER TABLE DISTRICT72 DROP CONSTRAINT DISTRICT72CKC;
ALTER TABLE DISTRICT72 ADD CONSTRAINT DISTRICT72CKC CHECK (D_W_ID
BETWEEN 36921 AND 37440);
SET INTEGRITY FOR DISTRICT72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT73 OFF;
ALTER TABLE DISTRICT73 DROP CONSTRAINT DISTRICT73CKC;
ALTER TABLE DISTRICT73 ADD CONSTRAINT DISTRICT73CKC CHECK (D_W_ID
BETWEEN 37441 AND 37960);
SET INTEGRITY FOR DISTRICT73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT74 OFF;
ALTER TABLE DISTRICT74 DROP CONSTRAINT DISTRICT74CKC;
ALTER TABLE DISTRICT74 ADD CONSTRAINT DISTRICT74CKC CHECK (D_W_ID
BETWEEN 37961 AND 38480);
SET INTEGRITY FOR DISTRICT74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT75 OFF;
ALTER TABLE DISTRICT75 DROP CONSTRAINT DISTRICT75CKC;
ALTER TABLE DISTRICT75 ADD CONSTRAINT DISTRICT75CKC CHECK (D_W_ID
BETWEEN 38481 AND 39000);
SET INTEGRITY FOR DISTRICT75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT76 OFF;
ALTER TABLE DISTRICT76 DROP CONSTRAINT DISTRICT76CKC;
ALTER TABLE DISTRICT76 ADD CONSTRAINT DISTRICT76CKC CHECK (D_W_ID
BETWEEN 39001 AND 39520);
SET INTEGRITY FOR DISTRICT76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT77 OFF;
ALTER TABLE DISTRICT77 DROP CONSTRAINT DISTRICT77CKC;
ALTER TABLE DISTRICT77 ADD CONSTRAINT DISTRICT77CKC CHECK (D_W_ID
BETWEEN 39521 AND 40040);
SET INTEGRITY FOR DISTRICT77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT78 OFF;
ALTER TABLE DISTRICT78 DROP CONSTRAINT DISTRICT78CKC;
ALTER TABLE DISTRICT78 ADD CONSTRAINT DISTRICT78CKC CHECK (D_W_ID
BETWEEN 40041 AND 40560);
SET INTEGRITY FOR DISTRICT78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT79 OFF;
ALTER TABLE DISTRICT79 DROP CONSTRAINT DISTRICT79CKC;
ALTER TABLE DISTRICT79 ADD CONSTRAINT DISTRICT79CKC CHECK (D_W_ID
BETWEEN 40561 AND 41080);
SET INTEGRITY FOR DISTRICT79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT80 OFF;
ALTER TABLE DISTRICT80 DROP CONSTRAINT DISTRICT80CKC;
ALTER TABLE DISTRICT80 ADD CONSTRAINT DISTRICT80CKC CHECK (D_W_ID >=
41081);
SET INTEGRITY FOR DISTRICT80 ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_NEW_ORDERA.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA1 OFF;
ALTER TABLE NEW_ORDERA1 DROP CONSTRAINT NEW_ORDERA1CKC;
ALTER TABLE NEW_ORDERA1 ADD CONSTRAINT NEW_ORDERA1CKC CHECK
((NO_W_ID BETWEEN 1 AND 520) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA2 OFF;
ALTER TABLE NEW_ORDERA2 DROP CONSTRAINT NEW_ORDERA2CKC;
ALTER TABLE NEW_ORDERA2 ADD CONSTRAINT NEW_ORDERA2CKC CHECK
((NO_W_ID BETWEEN 521 AND 1040) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA3 OFF;
ALTER TABLE NEW_ORDERA3 DROP CONSTRAINT NEW_ORDERA3CKC;
ALTER TABLE NEW_ORDERA3 ADD CONSTRAINT NEW_ORDERA3CKC CHECK
((NO_W_ID BETWEEN 1041 AND 1560) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA4 OFF;
ALTER TABLE NEW_ORDERA4 DROP CONSTRAINT NEW_ORDERA4CKC;
```

```
ALTER TABLE NEW_ORDERA4 ADD CONSTRAINT NEW_ORDERA4CKC CHECK
((NO_W_ID BETWEEN 1561 AND 2080) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA5 OFF;
ALTER TABLE NEW_ORDERA5 DROP CONSTRAINT NEW_ORDERA5CKC;
ALTER TABLE NEW_ORDERA5 ADD CONSTRAINT NEW_ORDERA5CKC CHECK
((NO_W_ID BETWEEN 2081 AND 2600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA6 OFF;
ALTER TABLE NEW_ORDERA6 DROP CONSTRAINT NEW_ORDERA6CKC;
ALTER TABLE NEW_ORDERA6 ADD CONSTRAINT NEW_ORDERA6CKC CHECK
((NO_W_ID BETWEEN 2601 AND 3120) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA7 OFF;
ALTER TABLE NEW_ORDERA7 DROP CONSTRAINT NEW_ORDERA7CKC;
ALTER TABLE NEW_ORDERA7 ADD CONSTRAINT NEW_ORDERA7CKC CHECK
((NO_W_ID BETWEEN 3121 AND 3640) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA8 OFF;
ALTER TABLE NEW_ORDERA8 DROP CONSTRAINT NEW_ORDERA8CKC;
ALTER TABLE NEW_ORDERA8 ADD CONSTRAINT NEW_ORDERA8CKC CHECK
((NO_W_ID BETWEEN 3641 AND 4160) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA9 OFF;
ALTER TABLE NEW_ORDERA9 DROP CONSTRAINT NEW_ORDERA9CKC;
ALTER TABLE NEW_ORDERA9 ADD CONSTRAINT NEW_ORDERA9CKC CHECK
((NO_W_ID BETWEEN 4161 AND 4680) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA10 OFF;
ALTER TABLE NEW_ORDERA10 DROP CONSTRAINT NEW_ORDERA10CKC;
ALTER TABLE NEW_ORDERA10 ADD CONSTRAINT NEW_ORDERA10CKC CHECK
((NO_W_ID BETWEEN 4681 AND 5200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA11 OFF;
ALTER TABLE NEW_ORDERA11 DROP CONSTRAINT NEW_ORDERA11CKC;
ALTER TABLE NEW_ORDERA11 ADD CONSTRAINT NEW_ORDERA11CKC CHECK
((NO_W_ID BETWEEN 5201 AND 5720) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA12 OFF;
ALTER TABLE NEW_ORDERA12 DROP CONSTRAINT NEW_ORDERA12CKC;
ALTER TABLE NEW_ORDERA12 ADD CONSTRAINT NEW_ORDERA12CKC CHECK
((NO_W_ID BETWEEN 5721 AND 6240) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA13 OFF;
ALTER TABLE NEW_ORDERA13 DROP CONSTRAINT NEW_ORDERA13CKC;
ALTER TABLE NEW_ORDERA13 ADD CONSTRAINT NEW_ORDERA13CKC CHECK
((NO_W_ID BETWEEN 6241 AND 6760) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA14 OFF;
ALTER TABLE NEW_ORDERA14 DROP CONSTRAINT NEW_ORDERA14CKC;
ALTER TABLE NEW_ORDERA14 ADD CONSTRAINT NEW_ORDERA14CKC CHECK
((NO_W_ID BETWEEN 6761 AND 7280) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA15 OFF;
ALTER TABLE NEW_ORDERA15 DROP CONSTRAINT NEW_ORDERA15CKC;
ALTER TABLE NEW_ORDERA15 ADD CONSTRAINT NEW_ORDERA15CKC CHECK
((NO_W_ID BETWEEN 7281 AND 7800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA16 OFF;
ALTER TABLE NEW_ORDERA16 DROP CONSTRAINT NEW_ORDERA16CKC;
ALTER TABLE NEW_ORDERA16 ADD CONSTRAINT NEW_ORDERA16CKC CHECK
((NO_W_ID BETWEEN 7801 AND 8320) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA17 OFF;
ALTER TABLE NEW_ORDERA17 DROP CONSTRAINT NEW_ORDERA17CKC;
ALTER TABLE NEW_ORDERA17 ADD CONSTRAINT NEW_ORDERA17CKC CHECK
((NO_W_ID BETWEEN 8321 AND 8840) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA18 OFF;
ALTER TABLE NEW_ORDERA18 DROP CONSTRAINT NEW_ORDERA18CKC;
ALTER TABLE NEW_ORDERA18 ADD CONSTRAINT NEW_ORDERA18CKC CHECK
((NO_W_ID BETWEEN 8841 AND 9360) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA19 OFF;
ALTER TABLE NEW_ORDERA19 DROP CONSTRAINT NEW_ORDERA19CKC;
ALTER TABLE NEW_ORDERA19 ADD CONSTRAINT NEW_ORDERA19CKC CHECK
((NO_W_ID BETWEEN 9361 AND 9880) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA20 OFF;
ALTER TABLE NEW_ORDERA20 DROP CONSTRAINT NEW_ORDERA20CKC;
ALTER TABLE NEW_ORDERA20 ADD CONSTRAINT NEW_ORDERA20CKC CHECK
((NO_W_ID BETWEEN 9881 AND 10400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA21 OFF;
ALTER TABLE NEW_ORDERA21 DROP CONSTRAINT NEW_ORDERA21CKC;
ALTER TABLE NEW_ORDERA21 ADD CONSTRAINT NEW_ORDERA21CKC CHECK
((NO_W_ID BETWEEN 10401 AND 10920) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA22 OFF;
ALTER TABLE NEW_ORDERA22 DROP CONSTRAINT NEW_ORDERA22CKC;
ALTER TABLE NEW_ORDERA22 ADD CONSTRAINT NEW_ORDERA22CKC CHECK
((NO_W_ID BETWEEN 10921 AND 11440) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA23 OFF;
ALTER TABLE NEW_ORDERA23 DROP CONSTRAINT NEW_ORDERA23CKC;
ALTER TABLE NEW_ORDERA23 ADD CONSTRAINT NEW_ORDERA23CKC CHECK
((NO_W_ID BETWEEN 11441 AND 11960) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA24 OFF;
ALTER TABLE NEW_ORDERA24 DROP CONSTRAINT NEW_ORDERA24CKC;
ALTER TABLE NEW_ORDERA24 ADD CONSTRAINT NEW_ORDERA24CKC CHECK
((NO_W_ID BETWEEN 11961 AND 12480) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA25 OFF;
ALTER TABLE NEW_ORDERA25 DROP CONSTRAINT NEW_ORDERA25CKC;
ALTER TABLE NEW_ORDERA25 ADD CONSTRAINT NEW_ORDERA25CKC CHECK
((NO_W_ID BETWEEN 12481 AND 13000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA26 OFF;
ALTER TABLE NEW_ORDERA26 DROP CONSTRAINT NEW_ORDERA26CKC;
ALTER TABLE NEW_ORDERA26 ADD CONSTRAINT NEW_ORDERA26CKC CHECK
((NO_W_ID BETWEEN 13001 AND 13520) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA27 OFF;
ALTER TABLE NEW_ORDERA27 DROP CONSTRAINT NEW_ORDERA27CKC;
ALTER TABLE NEW_ORDERA27 ADD CONSTRAINT NEW_ORDERA27CKC CHECK
((NO_W_ID BETWEEN 13521 AND 14040) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA28 OFF;
ALTER TABLE NEW_ORDERA28 DROP CONSTRAINT NEW_ORDERA28CKC;
ALTER TABLE NEW_ORDERA28 ADD CONSTRAINT NEW_ORDERA28CKC CHECK
((NO_W_ID BETWEEN 14041 AND 14560) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA29 OFF;
ALTER TABLE NEW_ORDERA29 DROP CONSTRAINT NEW_ORDERA29CKC;
ALTER TABLE NEW_ORDERA29 ADD CONSTRAINT NEW_ORDERA29CKC CHECK
((NO_W_ID BETWEEN 14561 AND 15080) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA30 OFF;
ALTER TABLE NEW_ORDERA30 DROP CONSTRAINT NEW_ORDERA30CKC;
ALTER TABLE NEW_ORDERA30 ADD CONSTRAINT NEW_ORDERA30CKC CHECK
((NO_W_ID BETWEEN 15081 AND 15600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA31 OFF;
ALTER TABLE NEW_ORDERA31 DROP CONSTRAINT NEW_ORDERA31CKC;
ALTER TABLE NEW_ORDERA31 ADD CONSTRAINT NEW_ORDERA31CKC CHECK
((NO_W_ID BETWEEN 15601 AND 16120) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA32 OFF;
ALTER TABLE NEW_ORDERA32 DROP CONSTRAINT NEW_ORDERA32CKC;
ALTER TABLE NEW_ORDERA32 ADD CONSTRAINT NEW_ORDERA32CKC CHECK
((NO_W_ID BETWEEN 16121 AND 16640) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA33 OFF;
ALTER TABLE NEW_ORDERA33 DROP CONSTRAINT NEW_ORDERA33CKC;
ALTER TABLE NEW_ORDERA33 ADD CONSTRAINT NEW_ORDERA33CKC CHECK
((NO_W_ID BETWEEN 16641 AND 17160) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA34 OFF;
ALTER TABLE NEW_ORDERA34 DROP CONSTRAINT NEW_ORDERA34CKC;
```

```
ALTER TABLE NEW_ORDERA34 ADD CONSTRAINT NEW_ORDERA34CKC CHECK ((NO_W_ID BETWEEN 17161 AND 17680) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA35 OFF;
ALTER TABLE NEW_ORDERA35 DROP CONSTRAINT NEW_ORDERA35CKC;
ALTER TABLE NEW_ORDERA35 ADD CONSTRAINT NEW_ORDERA35CKC CHECK ((NO_W_ID BETWEEN 17681 AND 18200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA36 OFF;
ALTER TABLE NEW_ORDERA36 DROP CONSTRAINT NEW_ORDERA36CKC;
ALTER TABLE NEW_ORDERA36 ADD CONSTRAINT NEW_ORDERA36CKC CHECK ((NO_W_ID BETWEEN 18201 AND 18720) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA37 OFF;
ALTER TABLE NEW_ORDERA37 DROP CONSTRAINT NEW_ORDERA37CKC;
ALTER TABLE NEW_ORDERA37 ADD CONSTRAINT NEW_ORDERA37CKC CHECK ((NO_W_ID BETWEEN 18721 AND 19240) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA38 OFF;
ALTER TABLE NEW_ORDERA38 DROP CONSTRAINT NEW_ORDERA38CKC;
ALTER TABLE NEW_ORDERA38 ADD CONSTRAINT NEW_ORDERA38CKC CHECK ((NO_W_ID BETWEEN 19241 AND 19760) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA39 OFF;
ALTER TABLE NEW_ORDERA39 DROP CONSTRAINT NEW_ORDERA39CKC;
ALTER TABLE NEW_ORDERA39 ADD CONSTRAINT NEW_ORDERA39CKC CHECK ((NO_W_ID BETWEEN 19761 AND 20280) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA40 OFF;
ALTER TABLE NEW_ORDERA40 DROP CONSTRAINT NEW_ORDERA40CKC;
ALTER TABLE NEW_ORDERA40 ADD CONSTRAINT NEW_ORDERA40CKC CHECK ((NO_W_ID BETWEEN 20281 AND 20800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA41 OFF;
ALTER TABLE NEW_ORDERA41 DROP CONSTRAINT NEW_ORDERA41CKC;
ALTER TABLE NEW_ORDERA41 ADD CONSTRAINT NEW_ORDERA41CKC CHECK ((NO_W_ID BETWEEN 20801 AND 21320) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA42 OFF;
ALTER TABLE NEW_ORDERA42 DROP CONSTRAINT NEW_ORDERA42CKC;
ALTER TABLE NEW_ORDERA42 ADD CONSTRAINT NEW_ORDERA42CKC CHECK ((NO_W_ID BETWEEN 21321 AND 21840) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA42 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA43 OFF;
ALTER TABLE NEW_ORDERA43 DROP CONSTRAINT NEW_ORDERA43CKC;
ALTER TABLE NEW_ORDERA43 ADD CONSTRAINT NEW_ORDERA43CKC CHECK ((NO_W_ID BETWEEN 21841 AND 22360) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA43 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA44 OFF;
ALTER TABLE NEW_ORDERA44 DROP CONSTRAINT NEW_ORDERA44CKC;
ALTER TABLE NEW_ORDERA44 ADD CONSTRAINT NEW_ORDERA44CKC CHECK ((NO_W_ID BETWEEN 22361 AND 22880) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA44 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA45 OFF;
ALTER TABLE NEW_ORDERA45 DROP CONSTRAINT NEW_ORDERA45CKC;
ALTER TABLE NEW_ORDERA45 ADD CONSTRAINT NEW_ORDERA45CKC CHECK ((NO_W_ID BETWEEN 22881 AND 23400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA46 OFF;
ALTER TABLE NEW_ORDERA46 DROP CONSTRAINT NEW_ORDERA46CKC;
ALTER TABLE NEW_ORDERA46 ADD CONSTRAINT NEW_ORDERA46CKC CHECK ((NO_W_ID BETWEEN 23401 AND 23920) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA47 OFF;
ALTER TABLE NEW_ORDERA47 DROP CONSTRAINT NEW_ORDERA47CKC;
ALTER TABLE NEW_ORDERA47 ADD CONSTRAINT NEW_ORDERA47CKC CHECK ((NO_W_ID BETWEEN 23921 AND 24440) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA48 OFF;
ALTER TABLE NEW_ORDERA48 DROP CONSTRAINT NEW_ORDERA48CKC;
ALTER TABLE NEW_ORDERA48 ADD CONSTRAINT NEW_ORDERA48CKC CHECK ((NO_W_ID BETWEEN 24441 AND 24960) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA49 OFF;
ALTER TABLE NEW_ORDERA49 DROP CONSTRAINT NEW_ORDERA49CKC;
ALTER TABLE NEW_ORDERA49 ADD CONSTRAINT NEW_ORDERA49CKC CHECK ((NO_W_ID BETWEEN 24961 AND 25480) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA50 OFF;
ALTER TABLE NEW_ORDERA50 DROP CONSTRAINT NEW_ORDERA50CKC;
ALTER TABLE NEW_ORDERA50 ADD CONSTRAINT NEW_ORDERA50CKC CHECK ((NO_W_ID BETWEEN 25481 AND 26000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA51 OFF;
ALTER TABLE NEW_ORDERA51 DROP CONSTRAINT NEW_ORDERA51CKC;
ALTER TABLE NEW_ORDERA51 ADD CONSTRAINT NEW_ORDERA51CKC CHECK ((NO_W_ID BETWEEN 26001 AND 26520) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA52 OFF;
ALTER TABLE NEW_ORDERA52 DROP CONSTRAINT NEW_ORDERA52CKC;
ALTER TABLE NEW_ORDERA52 ADD CONSTRAINT NEW_ORDERA52CKC CHECK ((NO_W_ID BETWEEN 26521 AND 27040) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA53 OFF;
ALTER TABLE NEW_ORDERA53 DROP CONSTRAINT NEW_ORDERA53CKC;
ALTER TABLE NEW_ORDERA53 ADD CONSTRAINT NEW_ORDERA53CKC CHECK ((NO_W_ID BETWEEN 27041 AND 27560) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA54 OFF;
ALTER TABLE NEW_ORDERA54 DROP CONSTRAINT NEW_ORDERA54CKC;
ALTER TABLE NEW_ORDERA54 ADD CONSTRAINT NEW_ORDERA54CKC CHECK ((NO_W_ID BETWEEN 27561 AND 28080) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA55 OFF;
ALTER TABLE NEW_ORDERA55 DROP CONSTRAINT NEW_ORDERA55CKC;
ALTER TABLE NEW_ORDERA55 ADD CONSTRAINT NEW_ORDERA55CKC CHECK ((NO_W_ID BETWEEN 28081 AND 28600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA56 OFF;
ALTER TABLE NEW_ORDERA56 DROP CONSTRAINT NEW_ORDERA56CKC;
ALTER TABLE NEW_ORDERA56 ADD CONSTRAINT NEW_ORDERA56CKC CHECK ((NO_W_ID BETWEEN 28601 AND 29120) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA57 OFF;
ALTER TABLE NEW_ORDERA57 DROP CONSTRAINT NEW_ORDERA57CKC;
ALTER TABLE NEW_ORDERA57 ADD CONSTRAINT NEW_ORDERA57CKC CHECK ((NO_W_ID BETWEEN 29121 AND 29640) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA58 OFF;
ALTER TABLE NEW_ORDERA58 DROP CONSTRAINT NEW_ORDERA58CKC;
ALTER TABLE NEW_ORDERA58 ADD CONSTRAINT NEW_ORDERA58CKC CHECK ((NO_W_ID BETWEEN 29641 AND 30160) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA59 OFF;
ALTER TABLE NEW_ORDERA59 DROP CONSTRAINT NEW_ORDERA59CKC;
ALTER TABLE NEW_ORDERA59 ADD CONSTRAINT NEW_ORDERA59CKC CHECK ((NO_W_ID BETWEEN 30161 AND 30680) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA60 OFF;
ALTER TABLE NEW_ORDERA60 DROP CONSTRAINT NEW_ORDERA60CKC;
ALTER TABLE NEW_ORDERA60 ADD CONSTRAINT NEW_ORDERA60CKC CHECK ((NO_W_ID BETWEEN 30681 AND 31200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA61 OFF;
ALTER TABLE NEW_ORDERA61 DROP CONSTRAINT NEW_ORDERA61CKC;
ALTER TABLE NEW_ORDERA61 ADD CONSTRAINT NEW_ORDERA61CKC CHECK ((NO_W_ID BETWEEN 31201 AND 31720) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA62 OFF;
ALTER TABLE NEW_ORDERA62 DROP CONSTRAINT NEW_ORDERA62CKC;
ALTER TABLE NEW_ORDERA62 ADD CONSTRAINT NEW_ORDERA62CKC CHECK ((NO_W_ID BETWEEN 31721 AND 32240) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA63 OFF;
ALTER TABLE NEW_ORDERA63 DROP CONSTRAINT NEW_ORDERA63CKC;
ALTER TABLE NEW_ORDERA63 ADD CONSTRAINT NEW_ORDERA63CKC CHECK ((NO_W_ID BETWEEN 32241 AND 32760) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA64 OFF;
ALTER TABLE NEW_ORDERA64 DROP CONSTRAINT NEW_ORDERA64CKC;
```

```
ALTER TABLE NEW_ORDERA64 ADD CONSTRAINT NEW_ORDERA64CKC CHECK
((NO_W_ID BETWEEN 32761 AND 33280) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA65 OFF;
ALTER TABLE NEW_ORDERA65 DROP CONSTRAINT NEW_ORDERA65CKC;
ALTER TABLE NEW_ORDERA65 ADD CONSTRAINT NEW_ORDERA65CKC CHECK
((NO_W_ID BETWEEN 33281 AND 33800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA66 OFF;
ALTER TABLE NEW_ORDERA66 DROP CONSTRAINT NEW_ORDERA66CKC;
ALTER TABLE NEW_ORDERA66 ADD CONSTRAINT NEW_ORDERA66CKC CHECK
((NO_W_ID BETWEEN 33801 AND 34320) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA67 OFF;
ALTER TABLE NEW_ORDERA67 DROP CONSTRAINT NEW_ORDERA67CKC;
ALTER TABLE NEW_ORDERA67 ADD CONSTRAINT NEW_ORDERA67CKC CHECK
((NO_W_ID BETWEEN 34321 AND 34840) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA68 OFF;
ALTER TABLE NEW_ORDERA68 DROP CONSTRAINT NEW_ORDERA68CKC;
ALTER TABLE NEW_ORDERA68 ADD CONSTRAINT NEW_ORDERA68CKC CHECK
((NO_W_ID BETWEEN 34841 AND 35360) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA69 OFF;
ALTER TABLE NEW_ORDERA69 DROP CONSTRAINT NEW_ORDERA69CKC;
ALTER TABLE NEW_ORDERA69 ADD CONSTRAINT NEW_ORDERA69CKC CHECK
((NO_W_ID BETWEEN 35361 AND 35880) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA70 OFF;
ALTER TABLE NEW_ORDERA70 DROP CONSTRAINT NEW_ORDERA70CKC;
ALTER TABLE NEW_ORDERA70 ADD CONSTRAINT NEW_ORDERA70CKC CHECK
((NO_W_ID BETWEEN 35881 AND 36400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA71 OFF;
ALTER TABLE NEW_ORDERA71 DROP CONSTRAINT NEW_ORDERA71CKC;
ALTER TABLE NEW_ORDERA71 ADD CONSTRAINT NEW_ORDERA71CKC CHECK
((NO_W_ID BETWEEN 36401 AND 36920) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA72 OFF;
ALTER TABLE NEW_ORDERA72 DROP CONSTRAINT NEW_ORDERA72CKC;
ALTER TABLE NEW_ORDERA72 ADD CONSTRAINT NEW_ORDERA72CKC CHECK
((NO_W_ID BETWEEN 36921 AND 37440) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA73 OFF;
ALTER TABLE NEW_ORDERA73 DROP CONSTRAINT NEW_ORDERA73CKC;
ALTER TABLE NEW_ORDERA73 ADD CONSTRAINT NEW_ORDERA73CKC CHECK
((NO_W_ID BETWEEN 37441 AND 37960) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA74 OFF;
ALTER TABLE NEW_ORDERA74 DROP CONSTRAINT NEW_ORDERA74CKC;
ALTER TABLE NEW_ORDERA74 ADD CONSTRAINT NEW_ORDERA74CKC CHECK
((NO_W_ID BETWEEN 37961 AND 38480) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA75 OFF;
ALTER TABLE NEW_ORDERA75 DROP CONSTRAINT NEW_ORDERA75CKC;
ALTER TABLE NEW_ORDERA75 ADD CONSTRAINT NEW_ORDERA75CKC CHECK
((NO_W_ID BETWEEN 38481 AND 39000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA76 OFF;
ALTER TABLE NEW_ORDERA76 DROP CONSTRAINT NEW_ORDERA76CKC;
ALTER TABLE NEW_ORDERA76 ADD CONSTRAINT NEW_ORDERA76CKC CHECK
((NO_W_ID BETWEEN 39001 AND 39520) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA77 OFF;
ALTER TABLE NEW_ORDERA77 DROP CONSTRAINT NEW_ORDERA77CKC;
ALTER TABLE NEW_ORDERA77 ADD CONSTRAINT NEW_ORDERA77CKC CHECK
((NO_W_ID BETWEEN 39521 AND 40040) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA78 OFF;
ALTER TABLE NEW_ORDERA78 DROP CONSTRAINT NEW_ORDERA78CKC;
ALTER TABLE NEW_ORDERA78 ADD CONSTRAINT NEW_ORDERA78CKC CHECK
((NO_W_ID BETWEEN 40041 AND 40560) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA79 OFF;
ALTER TABLE NEW_ORDERA79 DROP CONSTRAINT NEW_ORDERA79CKC;
ALTER TABLE NEW_ORDERA79 ADD CONSTRAINT NEW_ORDERA79CKC CHECK
((NO_W_ID BETWEEN 40561 AND 41080) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA80 OFF;
ALTER TABLE NEW_ORDERA80 DROP CONSTRAINT NEW_ORDERA80CKC;
ALTER TABLE NEW_ORDERA80 ADD CONSTRAINT NEW_ORDERA80CKC CHECK
((NO_W_ID >= 41081) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA80 ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_NEW_ORDERB.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB1 OFF;
ALTER TABLE NEW_ORDERB1 DROP CONSTRAINT NEW_ORDERB1CKC;
ALTER TABLE NEW_ORDERB1 ADD CONSTRAINT NEW_ORDERB1CKC CHECK
((NO_W_ID BETWEEN 1 AND 520) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB2 OFF;
ALTER TABLE NEW_ORDERB2 DROP CONSTRAINT NEW_ORDERB2CKC;
ALTER TABLE NEW_ORDERB2 ADD CONSTRAINT NEW_ORDERB2CKC CHECK
((NO_W_ID BETWEEN 521 AND 1040) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB3 OFF;
ALTER TABLE NEW_ORDERB3 DROP CONSTRAINT NEW_ORDERB3CKC;
ALTER TABLE NEW_ORDERB3 ADD CONSTRAINT NEW_ORDERB3CKC CHECK
((NO_W_ID BETWEEN 1041 AND 1560) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB4 OFF;
ALTER TABLE NEW_ORDERB4 DROP CONSTRAINT NEW_ORDERB4CKC;
ALTER TABLE NEW_ORDERB4 ADD CONSTRAINT NEW_ORDERB4CKC CHECK
((NO_W_ID BETWEEN 1561 AND 2080) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB5 OFF;
ALTER TABLE NEW_ORDERB5 DROP CONSTRAINT NEW_ORDERB5CKC;
ALTER TABLE NEW_ORDERB5 ADD CONSTRAINT NEW_ORDERB5CKC CHECK
((NO_W_ID BETWEEN 2081 AND 2600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB6 OFF;
ALTER TABLE NEW_ORDERB6 DROP CONSTRAINT NEW_ORDERB6CKC;
ALTER TABLE NEW_ORDERB6 ADD CONSTRAINT NEW_ORDERB6CKC CHECK
((NO_W_ID BETWEEN 2601 AND 3120) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB7 OFF;
ALTER TABLE NEW_ORDERB7 DROP CONSTRAINT NEW_ORDERB7CKC;
ALTER TABLE NEW_ORDERB7 ADD CONSTRAINT NEW_ORDERB7CKC CHECK
((NO_W_ID BETWEEN 3121 AND 3640) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB8 OFF;
ALTER TABLE NEW_ORDERB8 DROP CONSTRAINT NEW_ORDERB8CKC;
ALTER TABLE NEW_ORDERB8 ADD CONSTRAINT NEW_ORDERB8CKC CHECK
((NO_W_ID BETWEEN 3641 AND 4160) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB9 OFF;
ALTER TABLE NEW_ORDERB9 DROP CONSTRAINT NEW_ORDERB9CKC;
ALTER TABLE NEW_ORDERB9 ADD CONSTRAINT NEW_ORDERB9CKC CHECK
((NO_W_ID BETWEEN 4161 AND 4680) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB10 OFF;
ALTER TABLE NEW_ORDERB10 DROP CONSTRAINT NEW_ORDERB10CKC;
ALTER TABLE NEW_ORDERB10 ADD CONSTRAINT NEW_ORDERB10CKC CHECK
((NO_W_ID BETWEEN 4681 AND 5200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB11 OFF;
ALTER TABLE NEW_ORDERB11 DROP CONSTRAINT NEW_ORDERB11CKC;
ALTER TABLE NEW_ORDERB11 ADD CONSTRAINT NEW_ORDERB11CKC CHECK
((NO_W_ID BETWEEN 5201 AND 5720) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB12 OFF;
ALTER TABLE NEW_ORDERB12 DROP CONSTRAINT NEW_ORDERB12CKC;
ALTER TABLE NEW_ORDERB12 ADD CONSTRAINT NEW_ORDERB12CKC CHECK
((NO_W_ID BETWEEN 5721 AND 6240) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB13 OFF;
ALTER TABLE NEW_ORDERB13 DROP CONSTRAINT NEW_ORDERB13CKC;
ALTER TABLE NEW_ORDERB13 ADD CONSTRAINT NEW_ORDERB13CKC CHECK
((NO_W_ID BETWEEN 6241 AND 6760) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB14 OFF;
ALTER TABLE NEW_ORDERB14 DROP CONSTRAINT NEW_ORDERB14CKC;
ALTER TABLE NEW_ORDERB14 ADD CONSTRAINT NEW_ORDERB14CKC CHECK
((NO_W_ID BETWEEN 6761 AND 7280) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB15 OFF;
ALTER TABLE NEW_ORDERB15 DROP CONSTRAINT NEW_ORDERB15CKC;
ALTER TABLE NEW_ORDERB15 ADD CONSTRAINT NEW_ORDERB15CKC CHECK
((NO_W_ID BETWEEN 7281 AND 7800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB16 OFF;
ALTER TABLE NEW_ORDERB16 DROP CONSTRAINT NEW_ORDERB16CKC;
ALTER TABLE NEW_ORDERB16 ADD CONSTRAINT NEW_ORDERB16CKC CHECK
((NO_W_ID BETWEEN 7801 AND 8320) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB17 OFF;
ALTER TABLE NEW_ORDERB17 DROP CONSTRAINT NEW_ORDERB17CKC;
ALTER TABLE NEW_ORDERB17 ADD CONSTRAINT NEW_ORDERB17CKC CHECK
((NO_W_ID BETWEEN 8321 AND 8840) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB18 OFF;
ALTER TABLE NEW_ORDERB18 DROP CONSTRAINT NEW_ORDERB18CKC;
ALTER TABLE NEW_ORDERB18 ADD CONSTRAINT NEW_ORDERB18CKC CHECK
((NO_W_ID BETWEEN 8841 AND 9360) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB19 OFF;
ALTER TABLE NEW_ORDERB19 DROP CONSTRAINT NEW_ORDERB19CKC;
ALTER TABLE NEW_ORDERB19 ADD CONSTRAINT NEW_ORDERB19CKC CHECK
((NO_W_ID BETWEEN 9361 AND 9880) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB20 OFF;
ALTER TABLE NEW_ORDERB20 DROP CONSTRAINT NEW_ORDERB20CKC;
ALTER TABLE NEW_ORDERB20 ADD CONSTRAINT NEW_ORDERB20CKC CHECK
((NO_W_ID BETWEEN 9881 AND 10400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB21 OFF;
ALTER TABLE NEW_ORDERB21 DROP CONSTRAINT NEW_ORDERB21CKC;
ALTER TABLE NEW_ORDERB21 ADD CONSTRAINT NEW_ORDERB21CKC CHECK
((NO_W_ID BETWEEN 10401 AND 10920) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB22 OFF;
ALTER TABLE NEW_ORDERB22 DROP CONSTRAINT NEW_ORDERB22CKC;
ALTER TABLE NEW_ORDERB22 ADD CONSTRAINT NEW_ORDERB22CKC CHECK
((NO_W_ID BETWEEN 10921 AND 11440) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB23 OFF;
ALTER TABLE NEW_ORDERB23 DROP CONSTRAINT NEW_ORDERB23CKC;
ALTER TABLE NEW_ORDERB23 ADD CONSTRAINT NEW_ORDERB23CKC CHECK
((NO_W_ID BETWEEN 11441 AND 11960) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB24 OFF;
ALTER TABLE NEW_ORDERB24 DROP CONSTRAINT NEW_ORDERB24CKC;
ALTER TABLE NEW_ORDERB24 ADD CONSTRAINT NEW_ORDERB24CKC CHECK
((NO_W_ID BETWEEN 11961 AND 12480) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB25 OFF;
ALTER TABLE NEW_ORDERB25 DROP CONSTRAINT NEW_ORDERB25CKC;
ALTER TABLE NEW_ORDERB25 ADD CONSTRAINT NEW_ORDERB25CKC CHECK
((NO_W_ID BETWEEN 12481 AND 13000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB26 OFF;
ALTER TABLE NEW_ORDERB26 DROP CONSTRAINT NEW_ORDERB26CKC;
ALTER TABLE NEW_ORDERB26 ADD CONSTRAINT NEW_ORDERB26CKC CHECK
((NO_W_ID BETWEEN 13001 AND 13520) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB27 OFF;
ALTER TABLE NEW_ORDERB27 DROP CONSTRAINT NEW_ORDERB27CKC;
ALTER TABLE NEW_ORDERB27 ADD CONSTRAINT NEW_ORDERB27CKC CHECK
((NO_W_ID BETWEEN 13521 AND 14040) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB28 OFF;
ALTER TABLE NEW_ORDERB28 DROP CONSTRAINT NEW_ORDERB28CKC;
ALTER TABLE NEW_ORDERB28 ADD CONSTRAINT NEW_ORDERB28CKC CHECK
((NO_W_ID BETWEEN 14041 AND 14560) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB29 OFF;
ALTER TABLE NEW_ORDERB29 DROP CONSTRAINT NEW_ORDERB29CKC;
ALTER TABLE NEW_ORDERB29 ADD CONSTRAINT NEW_ORDERB29CKC CHECK
((NO_W_ID BETWEEN 14561 AND 15080) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB30 OFF;
ALTER TABLE NEW_ORDERB30 DROP CONSTRAINT NEW_ORDERB30CKC;
ALTER TABLE NEW_ORDERB30 ADD CONSTRAINT NEW_ORDERB30CKC CHECK
((NO_W_ID BETWEEN 15081 AND 15600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB31 OFF;
ALTER TABLE NEW_ORDERB31 DROP CONSTRAINT NEW_ORDERB31CKC;
ALTER TABLE NEW_ORDERB31 ADD CONSTRAINT NEW_ORDERB31CKC CHECK
((NO_W_ID BETWEEN 15601 AND 16120) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB32 OFF;
ALTER TABLE NEW_ORDERB32 DROP CONSTRAINT NEW_ORDERB32CKC;
ALTER TABLE NEW_ORDERB32 ADD CONSTRAINT NEW_ORDERB32CKC CHECK
((NO_W_ID BETWEEN 16121 AND 16640) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB33 OFF;
ALTER TABLE NEW_ORDERB33 DROP CONSTRAINT NEW_ORDERB33CKC;
ALTER TABLE NEW_ORDERB33 ADD CONSTRAINT NEW_ORDERB33CKC CHECK
((NO_W_ID BETWEEN 16641 AND 17160) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB33 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB34 OFF;
ALTER TABLE NEW_ORDERB34 DROP CONSTRAINT NEW_ORDERB34CKC;
ALTER TABLE NEW_ORDERB34 ADD CONSTRAINT NEW_ORDERB34CKC CHECK
((NO_W_ID BETWEEN 17161 AND 17680) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB35 OFF;
ALTER TABLE NEW_ORDERB35 DROP CONSTRAINT NEW_ORDERB35CKC;
ALTER TABLE NEW_ORDERB35 ADD CONSTRAINT NEW_ORDERB35CKC CHECK
((NO_W_ID BETWEEN 17681 AND 18200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB36 OFF;
ALTER TABLE NEW_ORDERB36 DROP CONSTRAINT NEW_ORDERB36CKC;
ALTER TABLE NEW_ORDERB36 ADD CONSTRAINT NEW_ORDERB36CKC CHECK
((NO_W_ID BETWEEN 18201 AND 18720) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB37 OFF;
ALTER TABLE NEW_ORDERB37 DROP CONSTRAINT NEW_ORDERB37CKC;
ALTER TABLE NEW_ORDERB37 ADD CONSTRAINT NEW_ORDERB37CKC CHECK
((NO_W_ID BETWEEN 18721 AND 19240) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB38 OFF;
ALTER TABLE NEW_ORDERB38 DROP CONSTRAINT NEW_ORDERB38CKC;
ALTER TABLE NEW_ORDERB38 ADD CONSTRAINT NEW_ORDERB38CKC CHECK
((NO_W_ID BETWEEN 19241 AND 19760) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB39 OFF;
ALTER TABLE NEW_ORDERB39 DROP CONSTRAINT NEW_ORDERB39CKC;
ALTER TABLE NEW_ORDERB39 ADD CONSTRAINT NEW_ORDERB39CKC CHECK
((NO_W_ID BETWEEN 19761 AND 20280) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB40 OFF;
ALTER TABLE NEW_ORDERB40 DROP CONSTRAINT NEW_ORDERB40CKC;
ALTER TABLE NEW_ORDERB40 ADD CONSTRAINT NEW_ORDERB40CKC CHECK
((NO_W_ID BETWEEN 20281 AND 20800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB41 OFF;
ALTER TABLE NEW_ORDERB41 DROP CONSTRAINT NEW_ORDERB41CKC;
ALTER TABLE NEW_ORDERB41 ADD CONSTRAINT NEW_ORDERB41CKC CHECK
((NO_W_ID BETWEEN 20801 AND 21320) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB42 OFF;
ALTER TABLE NEW_ORDERB42 DROP CONSTRAINT NEW_ORDERB42CKC;
ALTER TABLE NEW_ORDERB42 ADD CONSTRAINT NEW_ORDERB42CKC CHECK
((NO_W_ID BETWEEN 21321 AND 21840) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB42 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB43 OFF;
ALTER TABLE NEW_ORDERB43 DROP CONSTRAINT NEW_ORDERB43CKC;
ALTER TABLE NEW_ORDERB43 ADD CONSTRAINT NEW_ORDERB43CKC CHECK
((NO_W_ID BETWEEN 21841 AND 22360) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB43 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB44 OFF;
ALTER TABLE NEW_ORDERB44 DROP CONSTRAINT NEW_ORDERB44CKC;
ALTER TABLE NEW_ORDERB44 ADD CONSTRAINT NEW_ORDERB44CKC CHECK
((NO_W_ID BETWEEN 22361 AND 22880) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB44 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB45 OFF;
ALTER TABLE NEW_ORDERB45 DROP CONSTRAINT NEW_ORDERB45CKC;
ALTER TABLE NEW_ORDERB45 ADD CONSTRAINT NEW_ORDERB45CKC CHECK
((NO_W_ID BETWEEN 22881 AND 23400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB46 OFF;
ALTER TABLE NEW_ORDERB46 DROP CONSTRAINT NEW_ORDERB46CKC;
ALTER TABLE NEW_ORDERB46 ADD CONSTRAINT NEW_ORDERB46CKC CHECK
((NO_W_ID BETWEEN 23401 AND 23920) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB47 OFF;
ALTER TABLE NEW_ORDERB47 DROP CONSTRAINT NEW_ORDERB47CKC;
ALTER TABLE NEW_ORDERB47 ADD CONSTRAINT NEW_ORDERB47CKC CHECK
((NO_W_ID BETWEEN 23921 AND 24440) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB48 OFF;
ALTER TABLE NEW_ORDERB48 DROP CONSTRAINT NEW_ORDERB48CKC;
ALTER TABLE NEW_ORDERB48 ADD CONSTRAINT NEW_ORDERB48CKC CHECK
((NO_W_ID BETWEEN 24441 AND 24960) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB49 OFF;
ALTER TABLE NEW_ORDERB49 DROP CONSTRAINT NEW_ORDERB49CKC;
ALTER TABLE NEW_ORDERB49 ADD CONSTRAINT NEW_ORDERB49CKC CHECK
((NO_W_ID BETWEEN 24961 AND 25480) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB50 OFF;
ALTER TABLE NEW_ORDERB50 DROP CONSTRAINT NEW_ORDERB50CKC;
ALTER TABLE NEW_ORDERB50 ADD CONSTRAINT NEW_ORDERB50CKC CHECK
((NO_W_ID BETWEEN 25481 AND 26000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB51 OFF;
ALTER TABLE NEW_ORDERB51 DROP CONSTRAINT NEW_ORDERB51CKC;
ALTER TABLE NEW_ORDERB51 ADD CONSTRAINT NEW_ORDERB51CKC CHECK
((NO_W_ID BETWEEN 26001 AND 26520) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB52 OFF;
ALTER TABLE NEW_ORDERB52 DROP CONSTRAINT NEW_ORDERB52CKC;
ALTER TABLE NEW_ORDERB52 ADD CONSTRAINT NEW_ORDERB52CKC CHECK
((NO_W_ID BETWEEN 26521 AND 27040) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB53 OFF;
ALTER TABLE NEW_ORDERB53 DROP CONSTRAINT NEW_ORDERB53CKC;
ALTER TABLE NEW_ORDERB53 ADD CONSTRAINT NEW_ORDERB53CKC CHECK
((NO_W_ID BETWEEN 27041 AND 27560) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB53 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB54 OFF;
ALTER TABLE NEW_ORDERB54 DROP CONSTRAINT NEW_ORDERB54CKC;
ALTER TABLE NEW_ORDERB54 ADD CONSTRAINT NEW_ORDERB54CKC CHECK
((NO_W_ID BETWEEN 27561 AND 28080) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB55 OFF;
ALTER TABLE NEW_ORDERB55 DROP CONSTRAINT NEW_ORDERB55CKC;
ALTER TABLE NEW_ORDERB55 ADD CONSTRAINT NEW_ORDERB55CKC CHECK
((NO_W_ID BETWEEN 28081 AND 28600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB56 OFF;
ALTER TABLE NEW_ORDERB56 DROP CONSTRAINT NEW_ORDERB56CKC;
ALTER TABLE NEW_ORDERB56 ADD CONSTRAINT NEW_ORDERB56CKC CHECK
((NO_W_ID BETWEEN 28601 AND 29120) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB57 OFF;
ALTER TABLE NEW_ORDERB57 DROP CONSTRAINT NEW_ORDERB57CKC;
ALTER TABLE NEW_ORDERB57 ADD CONSTRAINT NEW_ORDERB57CKC CHECK
((NO_W_ID BETWEEN 29121 AND 29640) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB58 OFF;
ALTER TABLE NEW_ORDERB58 DROP CONSTRAINT NEW_ORDERB58CKC;
ALTER TABLE NEW_ORDERB58 ADD CONSTRAINT NEW_ORDERB58CKC CHECK
((NO_W_ID BETWEEN 29641 AND 30160) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB59 OFF;
ALTER TABLE NEW_ORDERB59 DROP CONSTRAINT NEW_ORDERB59CKC;
ALTER TABLE NEW_ORDERB59 ADD CONSTRAINT NEW_ORDERB59CKC CHECK
((NO_W_ID BETWEEN 30161 AND 30680) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB60 OFF;
ALTER TABLE NEW_ORDERB60 DROP CONSTRAINT NEW_ORDERB60CKC;
ALTER TABLE NEW_ORDERB60 ADD CONSTRAINT NEW_ORDERB60CKC CHECK
((NO_W_ID BETWEEN 30681 AND 31200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB61 OFF;
ALTER TABLE NEW_ORDERB61 DROP CONSTRAINT NEW_ORDERB61CKC;
ALTER TABLE NEW_ORDERB61 ADD CONSTRAINT NEW_ORDERB61CKC CHECK
((NO_W_ID BETWEEN 31201 AND 31720) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB62 OFF;
ALTER TABLE NEW_ORDERB62 DROP CONSTRAINT NEW_ORDERB62CKC;
ALTER TABLE NEW_ORDERB62 ADD CONSTRAINT NEW_ORDERB62CKC CHECK
((NO_W_ID BETWEEN 31721 AND 32240) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB63 OFF;
ALTER TABLE NEW_ORDERB63 DROP CONSTRAINT NEW_ORDERB63CKC;
ALTER TABLE NEW_ORDERB63 ADD CONSTRAINT NEW_ORDERB63CKC CHECK
((NO_W_ID BETWEEN 32241 AND 32760) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB63 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB64 OFF;
ALTER TABLE NEW_ORDERB64 DROP CONSTRAINT NEW_ORDERB64CKC;
ALTER TABLE NEW_ORDERB64 ADD CONSTRAINT NEW_ORDERB64CKC CHECK
((NO_W_ID BETWEEN 32761 AND 33280) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB65 OFF;
ALTER TABLE NEW_ORDERB65 DROP CONSTRAINT NEW_ORDERB65CKC;
ALTER TABLE NEW_ORDERB65 ADD CONSTRAINT NEW_ORDERB65CKC CHECK
((NO_W_ID BETWEEN 33281 AND 33800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB66 OFF;
ALTER TABLE NEW_ORDERB66 DROP CONSTRAINT NEW_ORDERB66CKC;
ALTER TABLE NEW_ORDERB66 ADD CONSTRAINT NEW_ORDERB66CKC CHECK
((NO_W_ID BETWEEN 33801 AND 34320) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB67 OFF;
ALTER TABLE NEW_ORDERB67 DROP CONSTRAINT NEW_ORDERB67CKC;
ALTER TABLE NEW_ORDERB67 ADD CONSTRAINT NEW_ORDERB67CKC CHECK
((NO_W_ID BETWEEN 34321 AND 34840) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB68 OFF;
ALTER TABLE NEW_ORDERB68 DROP CONSTRAINT NEW_ORDERB68CKC;
ALTER TABLE NEW_ORDERB68 ADD CONSTRAINT NEW_ORDERB68CKC CHECK
((NO_W_ID BETWEEN 34841 AND 35360) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB69 OFF;
ALTER TABLE NEW_ORDERB69 DROP CONSTRAINT NEW_ORDERB69CKC;
ALTER TABLE NEW_ORDERB69 ADD CONSTRAINT NEW_ORDERB69CKC CHECK
((NO_W_ID BETWEEN 35361 AND 35880) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB70 OFF;
ALTER TABLE NEW_ORDERB70 DROP CONSTRAINT NEW_ORDERB70CKC;
ALTER TABLE NEW_ORDERB70 ADD CONSTRAINT NEW_ORDERB70CKC CHECK
((NO_W_ID BETWEEN 35881 AND 36400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB71 OFF;
ALTER TABLE NEW_ORDERB71 DROP CONSTRAINT NEW_ORDERB71CKC;
ALTER TABLE NEW_ORDERB71 ADD CONSTRAINT NEW_ORDERB71CKC CHECK
((NO_W_ID BETWEEN 36401 AND 36920) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB72 OFF;
ALTER TABLE NEW_ORDERB72 DROP CONSTRAINT NEW_ORDERB72CKC;
ALTER TABLE NEW_ORDERB72 ADD CONSTRAINT NEW_ORDERB72CKC CHECK
((NO_W_ID BETWEEN 36921 AND 37440) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB73 OFF;
ALTER TABLE NEW_ORDERB73 DROP CONSTRAINT NEW_ORDERB73CKC;
ALTER TABLE NEW_ORDERB73 ADD CONSTRAINT NEW_ORDERB73CKC CHECK
((NO_W_ID BETWEEN 37441 AND 37960) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB73 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB74 OFF;
ALTER TABLE NEW_ORDERB74 DROP CONSTRAINT NEW_ORDERB74CKC;
ALTER TABLE NEW_ORDERB74 ADD CONSTRAINT NEW_ORDERB74CKC CHECK
((NO_W_ID BETWEEN 37961 AND 38480) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB75 OFF;
ALTER TABLE NEW_ORDERB75 DROP CONSTRAINT NEW_ORDERB75CKC;
ALTER TABLE NEW_ORDERB75 ADD CONSTRAINT NEW_ORDERB75CKC CHECK
((NO_W_ID BETWEEN 38481 AND 39000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB76 OFF;
ALTER TABLE NEW_ORDERB76 DROP CONSTRAINT NEW_ORDERB76CKC;
ALTER TABLE NEW_ORDERB76 ADD CONSTRAINT NEW_ORDERB76CKC CHECK
((NO_W_ID BETWEEN 39001 AND 39520) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB77 OFF;
ALTER TABLE NEW_ORDERB77 DROP CONSTRAINT NEW_ORDERB77CKC;
ALTER TABLE NEW_ORDERB77 ADD CONSTRAINT NEW_ORDERB77CKC CHECK
((NO_W_ID BETWEEN 39521 AND 40040) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB78 OFF;
ALTER TABLE NEW_ORDERB78 DROP CONSTRAINT NEW_ORDERB78CKC;
ALTER TABLE NEW_ORDERB78 ADD CONSTRAINT NEW_ORDERB78CKC CHECK
((NO_W_ID BETWEEN 40041 AND 40560) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB79 OFF;
ALTER TABLE NEW_ORDERB79 DROP CONSTRAINT NEW_ORDERB79CKC;
ALTER TABLE NEW_ORDERB79 ADD CONSTRAINT NEW_ORDERB79CKC CHECK
((NO_W_ID BETWEEN 40561 AND 41080) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB80 OFF;
ALTER TABLE NEW_ORDERB80 DROP CONSTRAINT NEW_ORDERB80CKC;
ALTER TABLE NEW_ORDERB80 ADD CONSTRAINT NEW_ORDERB80CKC CHECK
((NO_W_ID >= 41081) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB80 ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_ORDERS.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS1 OFF;
ALTER TABLE ORDERS1 DROP CONSTRAINT ORDERS1CKC;
ALTER TABLE ORDERS1 ADD CONSTRAINT ORDERS1CKC CHECK (O_W_ID
BETWEEN 1 AND 520);
SET INTEGRITY FOR ORDERS1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS2 OFF;
ALTER TABLE ORDERS2 DROP CONSTRAINT ORDERS2CKC;
ALTER TABLE ORDERS2 ADD CONSTRAINT ORDERS2CKC CHECK (O_W_ID
BETWEEN 521 AND 1040);
SET INTEGRITY FOR ORDERS2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS3 OFF;
ALTER TABLE ORDERS3 DROP CONSTRAINT ORDERS3CKC;
ALTER TABLE ORDERS3 ADD CONSTRAINT ORDERS3CKC CHECK (O_W_ID
BETWEEN 1041 AND 1560);
SET INTEGRITY FOR ORDERS3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS4 OFF;
ALTER TABLE ORDERS4 DROP CONSTRAINT ORDERS4CKC;
ALTER TABLE ORDERS4 ADD CONSTRAINT ORDERS4CKC CHECK (O_W_ID
BETWEEN 1561 AND 2080);
SET INTEGRITY FOR ORDERS4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS5 OFF;
ALTER TABLE ORDERS5 DROP CONSTRAINT ORDERS5CKC;
ALTER TABLE ORDERS5 ADD CONSTRAINT ORDERS5CKC CHECK (O_W_ID
BETWEEN 2081 AND 2600);
SET INTEGRITY FOR ORDERS5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS6 OFF;
ALTER TABLE ORDERS6 DROP CONSTRAINT ORDERS6CKC;
ALTER TABLE ORDERS6 ADD CONSTRAINT ORDERS6CKC CHECK (O_W_ID
BETWEEN 2601 AND 3120);
SET INTEGRITY FOR ORDERS6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS7 OFF;
ALTER TABLE ORDERS7 DROP CONSTRAINT ORDERS7CKC;
ALTER TABLE ORDERS7 ADD CONSTRAINT ORDERS7CKC CHECK (O_W_ID
BETWEEN 3121 AND 3640);
SET INTEGRITY FOR ORDERS7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS8 OFF;
ALTER TABLE ORDERS8 DROP CONSTRAINT ORDERS8CKC;
ALTER TABLE ORDERS8 ADD CONSTRAINT ORDERS8CKC CHECK (O_W_ID
BETWEEN 3641 AND 4160);
SET INTEGRITY FOR ORDERS8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS9 OFF;
ALTER TABLE ORDERS9 DROP CONSTRAINT ORDERS9CKC;
ALTER TABLE ORDERS9 ADD CONSTRAINT ORDERS9CKC CHECK (O_W_ID
BETWEEN 4161 AND 4680);
SET INTEGRITY FOR ORDERS9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS10 OFF;
ALTER TABLE ORDERS10 DROP CONSTRAINT ORDERS10CKC;
ALTER TABLE ORDERS10 ADD CONSTRAINT ORDERS10CKC CHECK (O_W_ID
BETWEEN 4681 AND 5200);
SET INTEGRITY FOR ORDERS10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS11 OFF;
ALTER TABLE ORDERS11 DROP CONSTRAINT ORDERS11CKC;
ALTER TABLE ORDERS11 ADD CONSTRAINT ORDERS11CKC CHECK (O_W_ID
BETWEEN 5201 AND 5720);
SET INTEGRITY FOR ORDERS11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS12 OFF;
ALTER TABLE ORDERS12 DROP CONSTRAINT ORDERS12CKC;
ALTER TABLE ORDERS12 ADD CONSTRAINT ORDERS12CKC CHECK (O_W_ID
BETWEEN 5721 AND 6240);
SET INTEGRITY FOR ORDERS12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS13 OFF;
ALTER TABLE ORDERS13 DROP CONSTRAINT ORDERS13CKC;
ALTER TABLE ORDERS13 ADD CONSTRAINT ORDERS13CKC CHECK (O_W_ID
BETWEEN 6241 AND 6760);
SET INTEGRITY FOR ORDERS13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS14 OFF;
ALTER TABLE ORDERS14 DROP CONSTRAINT ORDERS14CKC;
ALTER TABLE ORDERS14 ADD CONSTRAINT ORDERS14CKC CHECK (O_W_ID
BETWEEN 6761 AND 7280);
SET INTEGRITY FOR ORDERS14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS15 OFF;
ALTER TABLE ORDERS15 DROP CONSTRAINT ORDERS15CKC;
ALTER TABLE ORDERS15 ADD CONSTRAINT ORDERS15CKC CHECK (O_W_ID
BETWEEN 7281 AND 7800);
SET INTEGRITY FOR ORDERS15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS16 OFF;
ALTER TABLE ORDERS16 DROP CONSTRAINT ORDERS16CKC;
ALTER TABLE ORDERS16 ADD CONSTRAINT ORDERS16CKC CHECK (O_W_ID
BETWEEN 7801 AND 8320);
SET INTEGRITY FOR ORDERS16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS17 OFF;
ALTER TABLE ORDERS17 DROP CONSTRAINT ORDERS17CKC;
ALTER TABLE ORDERS17 ADD CONSTRAINT ORDERS17CKC CHECK (O_W_ID
BETWEEN 8321 AND 8840);
SET INTEGRITY FOR ORDERS17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS18 OFF;
ALTER TABLE ORDERS18 DROP CONSTRAINT ORDERS18CKC;
ALTER TABLE ORDERS18 ADD CONSTRAINT ORDERS18CKC CHECK (O_W_ID
BETWEEN 8841 AND 9360);
SET INTEGRITY FOR ORDERS18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS19 OFF;
ALTER TABLE ORDERS19 DROP CONSTRAINT ORDERS19CKC;
ALTER TABLE ORDERS19 ADD CONSTRAINT ORDERS19CKC CHECK (O_W_ID
BETWEEN 9361 AND 9880);
SET INTEGRITY FOR ORDERS19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS20 OFF;
ALTER TABLE ORDERS20 DROP CONSTRAINT ORDERS20CKC;
ALTER TABLE ORDERS20 ADD CONSTRAINT ORDERS20CKC CHECK (O_W_ID
BETWEEN 9881 AND 10400);
SET INTEGRITY FOR ORDERS20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS21 OFF;
ALTER TABLE ORDERS21 DROP CONSTRAINT ORDERS21CKC;
ALTER TABLE ORDERS21 ADD CONSTRAINT ORDERS21CKC CHECK (O_W_ID
BETWEEN 10401 AND 10920);
SET INTEGRITY FOR ORDERS21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS22 OFF;
ALTER TABLE ORDERS22 DROP CONSTRAINT ORDERS22CKC;
ALTER TABLE ORDERS22 ADD CONSTRAINT ORDERS22CKC CHECK (O_W_ID
BETWEEN 10921 AND 11440);
SET INTEGRITY FOR ORDERS22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS23 OFF;
ALTER TABLE ORDERS23 DROP CONSTRAINT ORDERS23CKC;
```

```
ALTER TABLE ORDERS23 ADD CONSTRAINT ORDERS23CKC CHECK (O_W_ID
BETWEEN 11441 AND 11960);
SET INTEGRITY FOR ORDERS23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS24 OFF;
ALTER TABLE ORDERS24 DROP CONSTRAINT ORDERS24CKC;
ALTER TABLE ORDERS24 ADD CONSTRAINT ORDERS24CKC CHECK (O_W_ID
BETWEEN 11961 AND 12480);
SET INTEGRITY FOR ORDERS24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS25 OFF;
ALTER TABLE ORDERS25 DROP CONSTRAINT ORDERS25CKC;
ALTER TABLE ORDERS25 ADD CONSTRAINT ORDERS25CKC CHECK (O_W_ID
BETWEEN 12481 AND 13000);
SET INTEGRITY FOR ORDERS25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS26 OFF;
ALTER TABLE ORDERS26 DROP CONSTRAINT ORDERS26CKC;
ALTER TABLE ORDERS26 ADD CONSTRAINT ORDERS26CKC CHECK (O_W_ID
BETWEEN 13001 AND 13520);
SET INTEGRITY FOR ORDERS26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS27 OFF;
ALTER TABLE ORDERS27 DROP CONSTRAINT ORDERS27CKC;
ALTER TABLE ORDERS27 ADD CONSTRAINT ORDERS27CKC CHECK (O_W_ID
BETWEEN 13521 AND 14040);
SET INTEGRITY FOR ORDERS27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS28 OFF;
ALTER TABLE ORDERS28 DROP CONSTRAINT ORDERS28CKC;
ALTER TABLE ORDERS28 ADD CONSTRAINT ORDERS28CKC CHECK (O_W_ID
BETWEEN 14041 AND 14560);
SET INTEGRITY FOR ORDERS28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS29 OFF;
ALTER TABLE ORDERS29 DROP CONSTRAINT ORDERS29CKC;
ALTER TABLE ORDERS29 ADD CONSTRAINT ORDERS29CKC CHECK (O_W_ID
BETWEEN 14561 AND 15080);
SET INTEGRITY FOR ORDERS29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS30 OFF;
ALTER TABLE ORDERS30 DROP CONSTRAINT ORDERS30CKC;
ALTER TABLE ORDERS30 ADD CONSTRAINT ORDERS30CKC CHECK (O_W_ID
BETWEEN 15081 AND 15600);
SET INTEGRITY FOR ORDERS30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS31 OFF;
ALTER TABLE ORDERS31 DROP CONSTRAINT ORDERS31CKC;
ALTER TABLE ORDERS31 ADD CONSTRAINT ORDERS31CKC CHECK (O_W_ID
BETWEEN 15601 AND 16120);
SET INTEGRITY FOR ORDERS31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS32 OFF;
ALTER TABLE ORDERS32 DROP CONSTRAINT ORDERS32CKC;
ALTER TABLE ORDERS32 ADD CONSTRAINT ORDERS32CKC CHECK (O_W_ID
BETWEEN 16121 AND 16640);
SET INTEGRITY FOR ORDERS32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS33 OFF;
ALTER TABLE ORDERS33 DROP CONSTRAINT ORDERS33CKC;
ALTER TABLE ORDERS33 ADD CONSTRAINT ORDERS33CKC CHECK (O_W_ID
BETWEEN 16641 AND 17160);
SET INTEGRITY FOR ORDERS33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS34 OFF;
ALTER TABLE ORDERS34 DROP CONSTRAINT ORDERS34CKC;
ALTER TABLE ORDERS34 ADD CONSTRAINT ORDERS34CKC CHECK (O_W_ID
BETWEEN 17161 AND 17680);
SET INTEGRITY FOR ORDERS34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS35 OFF;
ALTER TABLE ORDERS35 DROP CONSTRAINT ORDERS35CKC;
ALTER TABLE ORDERS35 ADD CONSTRAINT ORDERS35CKC CHECK (O_W_ID
BETWEEN 17681 AND 18200);
SET INTEGRITY FOR ORDERS35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS36 OFF;
ALTER TABLE ORDERS36 DROP CONSTRAINT ORDERS36CKC;
ALTER TABLE ORDERS36 ADD CONSTRAINT ORDERS36CKC CHECK (O_W_ID
BETWEEN 18201 AND 18720);
SET INTEGRITY FOR ORDERS36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS37 OFF;
ALTER TABLE ORDERS37 DROP CONSTRAINT ORDERS37CKC;
ALTER TABLE ORDERS37 ADD CONSTRAINT ORDERS37CKC CHECK (O_W_ID
BETWEEN 18721 AND 19240);
SET INTEGRITY FOR ORDERS37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS38 OFF;
ALTER TABLE ORDERS38 DROP CONSTRAINT ORDERS38CKC;
ALTER TABLE ORDERS38 ADD CONSTRAINT ORDERS38CKC CHECK (O_W_ID
BETWEEN 19241 AND 19760);
SET INTEGRITY FOR ORDERS38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS39 OFF;
ALTER TABLE ORDERS39 DROP CONSTRAINT ORDERS39CKC;
ALTER TABLE ORDERS39 ADD CONSTRAINT ORDERS39CKC CHECK (O_W_ID
BETWEEN 19761 AND 20280);
SET INTEGRITY FOR ORDERS39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS40 OFF;
ALTER TABLE ORDERS40 DROP CONSTRAINT ORDERS40CKC;
ALTER TABLE ORDERS40 ADD CONSTRAINT ORDERS40CKC CHECK (O_W_ID
BETWEEN 20281 AND 20800);
SET INTEGRITY FOR ORDERS40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS41 OFF;
ALTER TABLE ORDERS41 DROP CONSTRAINT ORDERS41CKC;
ALTER TABLE ORDERS41 ADD CONSTRAINT ORDERS41CKC CHECK (O_W_ID
BETWEEN 20801 AND 21320);
SET INTEGRITY FOR ORDERS41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS42 OFF;
ALTER TABLE ORDERS42 DROP CONSTRAINT ORDERS42CKC;
ALTER TABLE ORDERS42 ADD CONSTRAINT ORDERS42CKC CHECK (O_W_ID
BETWEEN 21321 AND 21840);
SET INTEGRITY FOR ORDERS42 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS43 OFF;
ALTER TABLE ORDERS43 DROP CONSTRAINT ORDERS43CKC;
ALTER TABLE ORDERS43 ADD CONSTRAINT ORDERS43CKC CHECK (O_W_ID
BETWEEN 21841 AND 22360);
SET INTEGRITY FOR ORDERS43 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS44 OFF;
ALTER TABLE ORDERS44 DROP CONSTRAINT ORDERS44CKC;
ALTER TABLE ORDERS44 ADD CONSTRAINT ORDERS44CKC CHECK (O_W_ID
BETWEEN 22361 AND 22880);
SET INTEGRITY FOR ORDERS44 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS45 OFF;
ALTER TABLE ORDERS45 DROP CONSTRAINT ORDERS45CKC;
ALTER TABLE ORDERS45 ADD CONSTRAINT ORDERS45CKC CHECK (O_W_ID
BETWEEN 22881 AND 23400);
SET INTEGRITY FOR ORDERS45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS46 OFF;
ALTER TABLE ORDERS46 DROP CONSTRAINT ORDERS46CKC;
ALTER TABLE ORDERS46 ADD CONSTRAINT ORDERS46CKC CHECK (O_W_ID
BETWEEN 23401 AND 23920);
SET INTEGRITY FOR ORDERS46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS47 OFF;
ALTER TABLE ORDERS47 DROP CONSTRAINT ORDERS47CKC;
ALTER TABLE ORDERS47 ADD CONSTRAINT ORDERS47CKC CHECK (O_W_ID
BETWEEN 23921 AND 24440);
SET INTEGRITY FOR ORDERS47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS48 OFF;
ALTER TABLE ORDERS48 DROP CONSTRAINT ORDERS48CKC;
ALTER TABLE ORDERS48 ADD CONSTRAINT ORDERS48CKC CHECK (O_W_ID
BETWEEN 24441 AND 24960);
SET INTEGRITY FOR ORDERS48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS49 OFF;
ALTER TABLE ORDERS49 DROP CONSTRAINT ORDERS49CKC;
ALTER TABLE ORDERS49 ADD CONSTRAINT ORDERS49CKC CHECK (O_W_ID
BETWEEN 24961 AND 25480);
SET INTEGRITY FOR ORDERS49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS50 OFF;
ALTER TABLE ORDERS50 DROP CONSTRAINT ORDERS50CKC;
ALTER TABLE ORDERS50 ADD CONSTRAINT ORDERS50CKC CHECK (O_W_ID
BETWEEN 25481 AND 26000);
SET INTEGRITY FOR ORDERS50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS51 OFF;
ALTER TABLE ORDERS51 DROP CONSTRAINT ORDERS51CKC;
ALTER TABLE ORDERS51 ADD CONSTRAINT ORDERS51CKC CHECK (O_W_ID
BETWEEN 26001 AND 26520);
SET INTEGRITY FOR ORDERS51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS52 OFF;
ALTER TABLE ORDERS52 DROP CONSTRAINT ORDERS52CKC;
ALTER TABLE ORDERS52 ADD CONSTRAINT ORDERS52CKC CHECK (O_W_ID
BETWEEN 26521 AND 27040);
SET INTEGRITY FOR ORDERS52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS53 OFF;
ALTER TABLE ORDERS53 DROP CONSTRAINT ORDERS53CKC;
```

```
ALTER TABLE ORDERS53 ADD CONSTRAINT ORDERS53CKC CHECK (O_W_ID
BETWEEN 27041 AND 27560);
SET INTEGRITY FOR ORDERS53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS54 OFF;
ALTER TABLE ORDERS54 DROP CONSTRAINT ORDERS54CKC;
ALTER TABLE ORDERS54 ADD CONSTRAINT ORDERS54CKC CHECK (O_W_ID
BETWEEN 27561 AND 28080);
SET INTEGRITY FOR ORDERS54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS55 OFF;
ALTER TABLE ORDERS55 DROP CONSTRAINT ORDERS55CKC;
ALTER TABLE ORDERS55 ADD CONSTRAINT ORDERS55CKC CHECK (O_W_ID
BETWEEN 28081 AND 28600);
SET INTEGRITY FOR ORDERS55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS56 OFF;
ALTER TABLE ORDERS56 DROP CONSTRAINT ORDERS56CKC;
ALTER TABLE ORDERS56 ADD CONSTRAINT ORDERS56CKC CHECK (O_W_ID
BETWEEN 28601 AND 29120);
SET INTEGRITY FOR ORDERS56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS57 OFF;
ALTER TABLE ORDERS57 DROP CONSTRAINT ORDERS57CKC;
ALTER TABLE ORDERS57 ADD CONSTRAINT ORDERS57CKC CHECK (O_W_ID
BETWEEN 29121 AND 29640);
SET INTEGRITY FOR ORDERS57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS58 OFF;
ALTER TABLE ORDERS58 DROP CONSTRAINT ORDERS58CKC;
ALTER TABLE ORDERS58 ADD CONSTRAINT ORDERS58CKC CHECK (O_W_ID
BETWEEN 29641 AND 30160);
SET INTEGRITY FOR ORDERS58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS59 OFF;
ALTER TABLE ORDERS59 DROP CONSTRAINT ORDERS59CKC;
ALTER TABLE ORDERS59 ADD CONSTRAINT ORDERS59CKC CHECK (O_W_ID
BETWEEN 30161 AND 30680);
SET INTEGRITY FOR ORDERS59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS60 OFF;
ALTER TABLE ORDERS60 DROP CONSTRAINT ORDERS60CKC;
ALTER TABLE ORDERS60 ADD CONSTRAINT ORDERS60CKC CHECK (O_W_ID
BETWEEN 30681 AND 31200);
SET INTEGRITY FOR ORDERS60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS61 OFF;
ALTER TABLE ORDERS61 DROP CONSTRAINT ORDERS61CKC;
ALTER TABLE ORDERS61 ADD CONSTRAINT ORDERS61CKC CHECK (O_W_ID
BETWEEN 31201 AND 31720);
SET INTEGRITY FOR ORDERS61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS62 OFF;
ALTER TABLE ORDERS62 DROP CONSTRAINT ORDERS62CKC;
ALTER TABLE ORDERS62 ADD CONSTRAINT ORDERS62CKC CHECK (O_W_ID
BETWEEN 31721 AND 32240);
SET INTEGRITY FOR ORDERS62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS63 OFF;
ALTER TABLE ORDERS63 DROP CONSTRAINT ORDERS63CKC;
ALTER TABLE ORDERS63 ADD CONSTRAINT ORDERS63CKC CHECK (O_W_ID
BETWEEN 32241 AND 32760);
SET INTEGRITY FOR ORDERS63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS64 OFF;
ALTER TABLE ORDERS64 DROP CONSTRAINT ORDERS64CKC;
ALTER TABLE ORDERS64 ADD CONSTRAINT ORDERS64CKC CHECK (O_W_ID
BETWEEN 32761 AND 33280);
SET INTEGRITY FOR ORDERS64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS65 OFF;
ALTER TABLE ORDERS65 DROP CONSTRAINT ORDERS65CKC;
ALTER TABLE ORDERS65 ADD CONSTRAINT ORDERS65CKC CHECK (O_W_ID
BETWEEN 33281 AND 33800);
SET INTEGRITY FOR ORDERS65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS66 OFF;
ALTER TABLE ORDERS66 DROP CONSTRAINT ORDERS66CKC;
ALTER TABLE ORDERS66 ADD CONSTRAINT ORDERS66CKC CHECK (O_W_ID
BETWEEN 33801 AND 34320);
SET INTEGRITY FOR ORDERS66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS67 OFF;
ALTER TABLE ORDERS67 DROP CONSTRAINT ORDERS67CKC;
ALTER TABLE ORDERS67 ADD CONSTRAINT ORDERS67CKC CHECK (O_W_ID
BETWEEN 34321 AND 34840);
SET INTEGRITY FOR ORDERS67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS68 OFF;
ALTER TABLE ORDERS68 DROP CONSTRAINT ORDERS68CKC;
ALTER TABLE ORDERS68 ADD CONSTRAINT ORDERS68CKC CHECK (O_W_ID
BETWEEN 34841 AND 35360);
SET INTEGRITY FOR ORDERS68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS69 OFF;
ALTER TABLE ORDERS69 DROP CONSTRAINT ORDERS69CKC;
ALTER TABLE ORDERS69 ADD CONSTRAINT ORDERS69CKC CHECK (O_W_ID
BETWEEN 35361 AND 35880);
SET INTEGRITY FOR ORDERS69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS70 OFF;
ALTER TABLE ORDERS70 DROP CONSTRAINT ORDERS70CKC;
ALTER TABLE ORDERS70 ADD CONSTRAINT ORDERS70CKC CHECK (O_W_ID
BETWEEN 35881 AND 36400);
SET INTEGRITY FOR ORDERS70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS71 OFF;
ALTER TABLE ORDERS71 DROP CONSTRAINT ORDERS71CKC;
ALTER TABLE ORDERS71 ADD CONSTRAINT ORDERS71CKC CHECK (O_W_ID
BETWEEN 36401 AND 36920);
SET INTEGRITY FOR ORDERS71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS72 OFF;
ALTER TABLE ORDERS72 DROP CONSTRAINT ORDERS72CKC;
ALTER TABLE ORDERS72 ADD CONSTRAINT ORDERS72CKC CHECK (O_W_ID
BETWEEN 36921 AND 37440);
SET INTEGRITY FOR ORDERS72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS73 OFF;
ALTER TABLE ORDERS73 DROP CONSTRAINT ORDERS73CKC;
ALTER TABLE ORDERS73 ADD CONSTRAINT ORDERS73CKC CHECK (O_W_ID
BETWEEN 37441 AND 37960);
SET INTEGRITY FOR ORDERS73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS74 OFF;
ALTER TABLE ORDERS74 DROP CONSTRAINT ORDERS74CKC;
ALTER TABLE ORDERS74 ADD CONSTRAINT ORDERS74CKC CHECK (O_W_ID
BETWEEN 37961 AND 38480);
SET INTEGRITY FOR ORDERS74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS75 OFF;
ALTER TABLE ORDERS75 DROP CONSTRAINT ORDERS75CKC;
ALTER TABLE ORDERS75 ADD CONSTRAINT ORDERS75CKC CHECK (O_W_ID
BETWEEN 38481 AND 39000);
SET INTEGRITY FOR ORDERS75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS76 OFF;
ALTER TABLE ORDERS76 DROP CONSTRAINT ORDERS76CKC;
ALTER TABLE ORDERS76 ADD CONSTRAINT ORDERS76CKC CHECK (O_W_ID
BETWEEN 39001 AND 39520);
SET INTEGRITY FOR ORDERS76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS77 OFF;
ALTER TABLE ORDERS77 DROP CONSTRAINT ORDERS77CKC;
ALTER TABLE ORDERS77 ADD CONSTRAINT ORDERS77CKC CHECK (O_W_ID
BETWEEN 39521 AND 40040);
SET INTEGRITY FOR ORDERS77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS78 OFF;
ALTER TABLE ORDERS78 DROP CONSTRAINT ORDERS78CKC;
ALTER TABLE ORDERS78 ADD CONSTRAINT ORDERS78CKC CHECK (O_W_ID
BETWEEN 40041 AND 40560);
SET INTEGRITY FOR ORDERS78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS79 OFF;
ALTER TABLE ORDERS79 DROP CONSTRAINT ORDERS79CKC;
ALTER TABLE ORDERS79 ADD CONSTRAINT ORDERS79CKC CHECK (O_W_ID
BETWEEN 40561 AND 41080);
SET INTEGRITY FOR ORDERS79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS80 OFF;
ALTER TABLE ORDERS80 DROP CONSTRAINT ORDERS80CKC;
ALTER TABLE ORDERS80 ADD CONSTRAINT ORDERS80CKC CHECK (O_W_ID >=
41081);
SET INTEGRITY FOR ORDERS80 ALL IMMEDIATE UNCHECKED;
connect reset;;
```

DDL/CRCONST_ORDER_LINE.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE1 OFF;
ALTER TABLE ORDER_LINE1 DROP CONSTRAINT ORDER_LINE1CKC;
ALTER TABLE ORDER_LINE1 ADD CONSTRAINT ORDER_LINE1CKC CHECK
(OL_W_ID BETWEEN 1 AND 520);
SET INTEGRITY FOR ORDER_LINE1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE2 OFF;
ALTER TABLE ORDER_LINE2 DROP CONSTRAINT ORDER_LINE2CKC;
ALTER TABLE ORDER_LINE2 ADD CONSTRAINT ORDER_LINE2CKC CHECK
(OL_W_ID BETWEEN 521 AND 1040);
SET INTEGRITY FOR ORDER_LINE2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE3 OFF;
ALTER TABLE ORDER_LINE3 DROP CONSTRAINT ORDER_LINE3CKC;
ALTER TABLE ORDER_LINE3 ADD CONSTRAINT ORDER_LINE3CKC CHECK
(OL_W_ID BETWEEN 1041 AND 1560);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE4 OFF;
ALTER TABLE ORDER_LINE4 DROP CONSTRAINT ORDER_LINE4CKC;
ALTER TABLE ORDER_LINE4 ADD CONSTRAINT ORDER_LINE4CKC CHECK
(OL_W_ID BETWEEN 1561 AND 2080);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE5 OFF;
ALTER TABLE ORDER_LINE5 DROP CONSTRAINT ORDER_LINE5CKC;
ALTER TABLE ORDER_LINE5 ADD CONSTRAINT ORDER_LINE5CKC CHECK
(OL_W_ID BETWEEN 2081 AND 2600);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE6 OFF;
ALTER TABLE ORDER_LINE6 DROP CONSTRAINT ORDER_LINE6CKC;
ALTER TABLE ORDER_LINE6 ADD CONSTRAINT ORDER_LINE6CKC CHECK
(OL_W_ID BETWEEN 2601 AND 3120);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE7 OFF;
ALTER TABLE ORDER_LINE7 DROP CONSTRAINT ORDER_LINE7CKC;
ALTER TABLE ORDER_LINE7 ADD CONSTRAINT ORDER_LINE7CKC CHECK
(OL_W_ID BETWEEN 3121 AND 3640);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE8 OFF;
ALTER TABLE ORDER_LINE8 DROP CONSTRAINT ORDER_LINE8CKC;
ALTER TABLE ORDER_LINE8 ADD CONSTRAINT ORDER_LINE8CKC CHECK
(OL_W_ID BETWEEN 3641 AND 4160);
SET INTEGRITY FOR ORDER_LINE8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE9 OFF;
ALTER TABLE ORDER_LINE9 DROP CONSTRAINT ORDER_LINE9CKC;
ALTER TABLE ORDER_LINE9 ADD CONSTRAINT ORDER_LINE9CKC CHECK
(OL_W_ID BETWEEN 4161 AND 4680);
SET INTEGRITY FOR ORDER_LINE9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE10 OFF;
ALTER TABLE ORDER_LINE10 DROP CONSTRAINT ORDER_LINE10CKC;
ALTER TABLE ORDER_LINE10 ADD CONSTRAINT ORDER_LINE10CKC CHECK
(OL_W_ID BETWEEN 4681 AND 5200);
SET INTEGRITY FOR ORDER_LINE10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE11 OFF;
ALTER TABLE ORDER_LINE11 DROP CONSTRAINT ORDER_LINE11CKC;
ALTER TABLE ORDER_LINE11 ADD CONSTRAINT ORDER_LINE11CKC CHECK
(OL_W_ID BETWEEN 5201 AND 5720);
SET INTEGRITY FOR ORDER_LINE11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE12 OFF;
ALTER TABLE ORDER_LINE12 DROP CONSTRAINT ORDER_LINE12CKC;
ALTER TABLE ORDER_LINE12 ADD CONSTRAINT ORDER_LINE12CKC CHECK
(OL_W_ID BETWEEN 5721 AND 6240);
SET INTEGRITY FOR ORDER_LINE12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE13 OFF;
ALTER TABLE ORDER_LINE13 DROP CONSTRAINT ORDER_LINE13CKC;
ALTER TABLE ORDER_LINE13 ADD CONSTRAINT ORDER_LINE13CKC CHECK
(OL_W_ID BETWEEN 6241 AND 6760);
SET INTEGRITY FOR ORDER_LINE13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE14 OFF;
ALTER TABLE ORDER_LINE14 DROP CONSTRAINT ORDER_LINE14CKC;
ALTER TABLE ORDER_LINE14 ADD CONSTRAINT ORDER_LINE14CKC CHECK
(OL_W_ID BETWEEN 6761 AND 7280);
SET INTEGRITY FOR ORDER_LINE14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE15 OFF;
ALTER TABLE ORDER_LINE15 DROP CONSTRAINT ORDER_LINE15CKC;
ALTER TABLE ORDER_LINE15 ADD CONSTRAINT ORDER_LINE15CKC CHECK
(OL_W_ID BETWEEN 7281 AND 7800);
SET INTEGRITY FOR ORDER_LINE15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE16 OFF;
ALTER TABLE ORDER_LINE16 DROP CONSTRAINT ORDER_LINE16CKC;
ALTER TABLE ORDER_LINE16 ADD CONSTRAINT ORDER_LINE16CKC CHECK
(OL_W_ID BETWEEN 7801 AND 8320);
SET INTEGRITY FOR ORDER_LINE16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE17 OFF;
ALTER TABLE ORDER_LINE17 DROP CONSTRAINT ORDER_LINE17CKC;
ALTER TABLE ORDER_LINE17 ADD CONSTRAINT ORDER_LINE17CKC CHECK
(OL_W_ID BETWEEN 8321 AND 8840);
SET INTEGRITY FOR ORDER_LINE17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE18 OFF;
ALTER TABLE ORDER_LINE18 DROP CONSTRAINT ORDER_LINE18CKC;
ALTER TABLE ORDER_LINE18 ADD CONSTRAINT ORDER_LINE18CKC CHECK
(OL_W_ID BETWEEN 8841 AND 9360);
SET INTEGRITY FOR ORDER_LINE18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE19 OFF;
ALTER TABLE ORDER_LINE19 DROP CONSTRAINT ORDER_LINE19CKC;
ALTER TABLE ORDER_LINE19 ADD CONSTRAINT ORDER_LINE19CKC CHECK
(OL_W_ID BETWEEN 9361 AND 9880);
SET INTEGRITY FOR ORDER_LINE19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE20 OFF;
ALTER TABLE ORDER_LINE20 DROP CONSTRAINT ORDER_LINE20CKC;
ALTER TABLE ORDER_LINE20 ADD CONSTRAINT ORDER_LINE20CKC CHECK
(OL_W_ID BETWEEN 9881 AND 10400);
SET INTEGRITY FOR ORDER_LINE20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE21 OFF;
ALTER TABLE ORDER_LINE21 DROP CONSTRAINT ORDER_LINE21CKC;
ALTER TABLE ORDER_LINE21 ADD CONSTRAINT ORDER_LINE21CKC CHECK
(OL_W_ID BETWEEN 10401 AND 10920);
SET INTEGRITY FOR ORDER_LINE21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE22 OFF;
ALTER TABLE ORDER_LINE22 DROP CONSTRAINT ORDER_LINE22CKC;
ALTER TABLE ORDER_LINE22 ADD CONSTRAINT ORDER_LINE22CKC CHECK
(OL_W_ID BETWEEN 10921 AND 11440);
SET INTEGRITY FOR ORDER_LINE22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE23 OFF;
ALTER TABLE ORDER_LINE23 DROP CONSTRAINT ORDER_LINE23CKC;
ALTER TABLE ORDER_LINE23 ADD CONSTRAINT ORDER_LINE23CKC CHECK
(OL_W_ID BETWEEN 11441 AND 11960);
SET INTEGRITY FOR ORDER_LINE23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE24 OFF;
ALTER TABLE ORDER_LINE24 DROP CONSTRAINT ORDER_LINE24CKC;
ALTER TABLE ORDER_LINE24 ADD CONSTRAINT ORDER_LINE24CKC CHECK
(OL_W_ID BETWEEN 11961 AND 12480);
SET INTEGRITY FOR ORDER_LINE24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE25 OFF;
ALTER TABLE ORDER_LINE25 DROP CONSTRAINT ORDER_LINE25CKC;
ALTER TABLE ORDER_LINE25 ADD CONSTRAINT ORDER_LINE25CKC CHECK
(OL_W_ID BETWEEN 12481 AND 13000);
SET INTEGRITY FOR ORDER_LINE25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE26 OFF;
ALTER TABLE ORDER_LINE26 DROP CONSTRAINT ORDER_LINE26CKC;
ALTER TABLE ORDER_LINE26 ADD CONSTRAINT ORDER_LINE26CKC CHECK
(OL_W_ID BETWEEN 13001 AND 13520);
SET INTEGRITY FOR ORDER_LINE26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE27 OFF;
ALTER TABLE ORDER_LINE27 DROP CONSTRAINT ORDER_LINE27CKC;
ALTER TABLE ORDER_LINE27 ADD CONSTRAINT ORDER_LINE27CKC CHECK
(OL_W_ID BETWEEN 13521 AND 14040);
SET INTEGRITY FOR ORDER_LINE27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE28 OFF;
ALTER TABLE ORDER_LINE28 DROP CONSTRAINT ORDER_LINE28CKC;
ALTER TABLE ORDER_LINE28 ADD CONSTRAINT ORDER_LINE28CKC CHECK
(OL_W_ID BETWEEN 14041 AND 14560);
SET INTEGRITY FOR ORDER_LINE28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE29 OFF;
ALTER TABLE ORDER_LINE29 DROP CONSTRAINT ORDER_LINE29CKC;
ALTER TABLE ORDER_LINE29 ADD CONSTRAINT ORDER_LINE29CKC CHECK
(OL_W_ID BETWEEN 14561 AND 15080);
SET INTEGRITY FOR ORDER_LINE29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE30 OFF;
ALTER TABLE ORDER_LINE30 DROP CONSTRAINT ORDER_LINE30CKC;
ALTER TABLE ORDER_LINE30 ADD CONSTRAINT ORDER_LINE30CKC CHECK
(OL_W_ID BETWEEN 15081 AND 15600);
SET INTEGRITY FOR ORDER_LINE30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE31 OFF;
ALTER TABLE ORDER_LINE31 DROP CONSTRAINT ORDER_LINE31CKC;
ALTER TABLE ORDER_LINE31 ADD CONSTRAINT ORDER_LINE31CKC CHECK
(OL_W_ID BETWEEN 15601 AND 16120);
SET INTEGRITY FOR ORDER_LINE31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE32 OFF;
ALTER TABLE ORDER_LINE32 DROP CONSTRAINT ORDER_LINE32CKC;
ALTER TABLE ORDER_LINE32 ADD CONSTRAINT ORDER_LINE32CKC CHECK
(OL_W_ID BETWEEN 16121 AND 16640);
SET INTEGRITY FOR ORDER_LINE32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE33 OFF;
ALTER TABLE ORDER_LINE33 DROP CONSTRAINT ORDER_LINE33CKC;
ALTER TABLE ORDER_LINE33 ADD CONSTRAINT ORDER_LINE33CKC CHECK
(OL_W_ID BETWEEN 16641 AND 17160);
SET INTEGRITY FOR ORDER_LINE33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE34 OFF;
ALTER TABLE ORDER_LINE34 DROP CONSTRAINT ORDER_LINE34CKC;
ALTER TABLE ORDER_LINE34 ADD CONSTRAINT ORDER_LINE34CKC CHECK
(OL_W_ID BETWEEN 17161 AND 17680);
SET INTEGRITY FOR ORDER_LINE34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE35 OFF;
ALTER TABLE ORDER_LINE35 DROP CONSTRAINT ORDER_LINE35CKC;
ALTER TABLE ORDER_LINE35 ADD CONSTRAINT ORDER_LINE35CKC CHECK
(OL_W_ID BETWEEN 17681 AND 18200);
SET INTEGRITY FOR ORDER_LINE35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE36 OFF;
ALTER TABLE ORDER_LINE36 DROP CONSTRAINT ORDER_LINE36CKC;
ALTER TABLE ORDER_LINE36 ADD CONSTRAINT ORDER_LINE36CKC CHECK
(OL_W_ID BETWEEN 18201 AND 18720);
SET INTEGRITY FOR ORDER_LINE36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE37 OFF;
ALTER TABLE ORDER_LINE37 DROP CONSTRAINT ORDER_LINE37CKC;
ALTER TABLE ORDER_LINE37 ADD CONSTRAINT ORDER_LINE37CKC CHECK
(OL_W_ID BETWEEN 18721 AND 19240);
SET INTEGRITY FOR ORDER_LINE37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE38 OFF;
ALTER TABLE ORDER_LINE38 DROP CONSTRAINT ORDER_LINE38CKC;
ALTER TABLE ORDER_LINE38 ADD CONSTRAINT ORDER_LINE38CKC CHECK
(OL_W_ID BETWEEN 19241 AND 19760);
SET INTEGRITY FOR ORDER_LINE38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE39 OFF;
ALTER TABLE ORDER_LINE39 DROP CONSTRAINT ORDER_LINE39CKC;
ALTER TABLE ORDER_LINE39 ADD CONSTRAINT ORDER_LINE39CKC CHECK
(OL_W_ID BETWEEN 19761 AND 20280);
SET INTEGRITY FOR ORDER_LINE39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE40 OFF;
ALTER TABLE ORDER_LINE40 DROP CONSTRAINT ORDER_LINE40CKC;
ALTER TABLE ORDER_LINE40 ADD CONSTRAINT ORDER_LINE40CKC CHECK
(OL_W_ID BETWEEN 20281 AND 20800);
SET INTEGRITY FOR ORDER_LINE40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE41 OFF;
ALTER TABLE ORDER_LINE41 DROP CONSTRAINT ORDER_LINE41CKC;
ALTER TABLE ORDER_LINE41 ADD CONSTRAINT ORDER_LINE41CKC CHECK
(OL_W_ID BETWEEN 20801 AND 21320);
SET INTEGRITY FOR ORDER_LINE41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE42 OFF;
ALTER TABLE ORDER_LINE42 DROP CONSTRAINT ORDER_LINE42CKC;
ALTER TABLE ORDER_LINE42 ADD CONSTRAINT ORDER_LINE42CKC CHECK
(OL_W_ID BETWEEN 21321 AND 21840);
SET INTEGRITY FOR ORDER_LINE42 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE43 OFF;
ALTER TABLE ORDER_LINE43 DROP CONSTRAINT ORDER_LINE43CKC;
ALTER TABLE ORDER_LINE43 ADD CONSTRAINT ORDER_LINE43CKC CHECK
(OL_W_ID BETWEEN 21841 AND 22360);
SET INTEGRITY FOR ORDER_LINE43 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE44 OFF;
ALTER TABLE ORDER_LINE44 DROP CONSTRAINT ORDER_LINE44CKC;
ALTER TABLE ORDER_LINE44 ADD CONSTRAINT ORDER_LINE44CKC CHECK
(OL_W_ID BETWEEN 22361 AND 22880);
SET INTEGRITY FOR ORDER_LINE44 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE45 OFF;
ALTER TABLE ORDER_LINE45 DROP CONSTRAINT ORDER_LINE45CKC;
ALTER TABLE ORDER_LINE45 ADD CONSTRAINT ORDER_LINE45CKC CHECK
(OL_W_ID BETWEEN 22881 AND 23400);
SET INTEGRITY FOR ORDER_LINE45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE46 OFF;
ALTER TABLE ORDER_LINE46 DROP CONSTRAINT ORDER_LINE46CKC;
ALTER TABLE ORDER_LINE46 ADD CONSTRAINT ORDER_LINE46CKC CHECK
(OL_W_ID BETWEEN 23401 AND 23920);
SET INTEGRITY FOR ORDER_LINE46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE47 OFF;
ALTER TABLE ORDER_LINE47 DROP CONSTRAINT ORDER_LINE47CKC;
ALTER TABLE ORDER_LINE47 ADD CONSTRAINT ORDER_LINE47CKC CHECK
(OL_W_ID BETWEEN 23921 AND 24440);
SET INTEGRITY FOR ORDER_LINE47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE48 OFF;
ALTER TABLE ORDER_LINE48 DROP CONSTRAINT ORDER_LINE48CKC;
ALTER TABLE ORDER_LINE48 ADD CONSTRAINT ORDER_LINE48CKC CHECK
(OL_W_ID BETWEEN 24441 AND 24960);
SET INTEGRITY FOR ORDER_LINE48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE49 OFF;
ALTER TABLE ORDER_LINE49 DROP CONSTRAINT ORDER_LINE49CKC;
ALTER TABLE ORDER_LINE49 ADD CONSTRAINT ORDER_LINE49CKC CHECK
(OL_W_ID BETWEEN 24961 AND 25480);
SET INTEGRITY FOR ORDER_LINE49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE50 OFF;
ALTER TABLE ORDER_LINE50 DROP CONSTRAINT ORDER_LINE50CKC;
ALTER TABLE ORDER_LINE50 ADD CONSTRAINT ORDER_LINE50CKC CHECK
(OL_W_ID BETWEEN 25481 AND 26000);
SET INTEGRITY FOR ORDER_LINE50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE51 OFF;
ALTER TABLE ORDER_LINE51 DROP CONSTRAINT ORDER_LINE51CKC;
ALTER TABLE ORDER_LINE51 ADD CONSTRAINT ORDER_LINE51CKC CHECK
(OL_W_ID BETWEEN 26001 AND 26520);
SET INTEGRITY FOR ORDER_LINE51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE52 OFF;
ALTER TABLE ORDER_LINE52 DROP CONSTRAINT ORDER_LINE52CKC;
ALTER TABLE ORDER_LINE52 ADD CONSTRAINT ORDER_LINE52CKC CHECK
(OL_W_ID BETWEEN 26521 AND 27040);
SET INTEGRITY FOR ORDER_LINE52 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE53 OFF;
ALTER TABLE ORDER_LINE53 DROP CONSTRAINT ORDER_LINE53CKC;
ALTER TABLE ORDER_LINE53 ADD CONSTRAINT ORDER_LINE53CKC CHECK
(OL_W_ID BETWEEN 27041 AND 27560);
SET INTEGRITY FOR ORDER_LINE53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE54 OFF;
ALTER TABLE ORDER_LINE54 DROP CONSTRAINT ORDER_LINE54CKC;
ALTER TABLE ORDER_LINE54 ADD CONSTRAINT ORDER_LINE54CKC CHECK
(OL_W_ID BETWEEN 27561 AND 28080);
SET INTEGRITY FOR ORDER_LINE54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE55 OFF;
ALTER TABLE ORDER_LINE55 DROP CONSTRAINT ORDER_LINE55CKC;
ALTER TABLE ORDER_LINE55 ADD CONSTRAINT ORDER_LINE55CKC CHECK
(OL_W_ID BETWEEN 28081 AND 28600);
SET INTEGRITY FOR ORDER_LINE55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE56 OFF;
ALTER TABLE ORDER_LINE56 DROP CONSTRAINT ORDER_LINE56CKC;
ALTER TABLE ORDER_LINE56 ADD CONSTRAINT ORDER_LINE56CKC CHECK
(OL_W_ID BETWEEN 28601 AND 29120);
SET INTEGRITY FOR ORDER_LINE56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE57 OFF;
ALTER TABLE ORDER_LINE57 DROP CONSTRAINT ORDER_LINE57CKC;
ALTER TABLE ORDER_LINE57 ADD CONSTRAINT ORDER_LINE57CKC CHECK
(OL_W_ID BETWEEN 29121 AND 29640);
SET INTEGRITY FOR ORDER_LINE57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE58 OFF;
ALTER TABLE ORDER_LINE58 DROP CONSTRAINT ORDER_LINE58CKC;
ALTER TABLE ORDER_LINE58 ADD CONSTRAINT ORDER_LINE58CKC CHECK
(OL_W_ID BETWEEN 29641 AND 30160);
SET INTEGRITY FOR ORDER_LINE58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE59 OFF;
ALTER TABLE ORDER_LINE59 DROP CONSTRAINT ORDER_LINE59CKC;
ALTER TABLE ORDER_LINE59 ADD CONSTRAINT ORDER_LINE59CKC CHECK
(OL_W_ID BETWEEN 30161 AND 30680);
SET INTEGRITY FOR ORDER_LINE59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE60 OFF;
ALTER TABLE ORDER_LINE60 DROP CONSTRAINT ORDER_LINE60CKC;
ALTER TABLE ORDER_LINE60 ADD CONSTRAINT ORDER_LINE60CKC CHECK
(OL_W_ID BETWEEN 30681 AND 31200);
SET INTEGRITY FOR ORDER_LINE60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE61 OFF;
ALTER TABLE ORDER_LINE61 DROP CONSTRAINT ORDER_LINE61CKC;
ALTER TABLE ORDER_LINE61 ADD CONSTRAINT ORDER_LINE61CKC CHECK
(OL_W_ID BETWEEN 31201 AND 31720);
SET INTEGRITY FOR ORDER_LINE61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE62 OFF;
ALTER TABLE ORDER_LINE62 DROP CONSTRAINT ORDER_LINE62CKC;
ALTER TABLE ORDER_LINE62 ADD CONSTRAINT ORDER_LINE62CKC CHECK
(OL_W_ID BETWEEN 31721 AND 32240);
SET INTEGRITY FOR ORDER_LINE62 ALL IMMEDIATE UNCHECKED;
connect reset;
```

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE63 OFF;
ALTER TABLE ORDER_LINE63 DROP CONSTRAINT ORDER_LINE63CKC;
ALTER TABLE ORDER_LINE63 ADD CONSTRAINT ORDER_LINE63CKC CHECK
(OL_W_ID BETWEEN 32241 AND 32760);
SET INTEGRITY FOR ORDER_LINE63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE64 OFF;
ALTER TABLE ORDER_LINE64 DROP CONSTRAINT ORDER_LINE64CKC;
ALTER TABLE ORDER_LINE64 ADD CONSTRAINT ORDER_LINE64CKC CHECK
(OL_W_ID BETWEEN 32761 AND 33280);
SET INTEGRITY FOR ORDER_LINE64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE65 OFF;
ALTER TABLE ORDER_LINE65 DROP CONSTRAINT ORDER_LINE65CKC;
ALTER TABLE ORDER_LINE65 ADD CONSTRAINT ORDER_LINE65CKC CHECK
(OL_W_ID BETWEEN 33281 AND 33800);
SET INTEGRITY FOR ORDER_LINE65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE66 OFF;
ALTER TABLE ORDER_LINE66 DROP CONSTRAINT ORDER_LINE66CKC;
ALTER TABLE ORDER_LINE66 ADD CONSTRAINT ORDER_LINE66CKC CHECK
(OL_W_ID BETWEEN 33801 AND 34320);
SET INTEGRITY FOR ORDER_LINE66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE67 OFF;
ALTER TABLE ORDER_LINE67 DROP CONSTRAINT ORDER_LINE67CKC;
ALTER TABLE ORDER_LINE67 ADD CONSTRAINT ORDER_LINE67CKC CHECK
(OL_W_ID BETWEEN 34321 AND 34840);
SET INTEGRITY FOR ORDER_LINE67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE68 OFF;
ALTER TABLE ORDER_LINE68 DROP CONSTRAINT ORDER_LINE68CKC;
ALTER TABLE ORDER_LINE68 ADD CONSTRAINT ORDER_LINE68CKC CHECK
(OL_W_ID BETWEEN 34841 AND 35360);
SET INTEGRITY FOR ORDER_LINE68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE69 OFF;
ALTER TABLE ORDER_LINE69 DROP CONSTRAINT ORDER_LINE69CKC;
ALTER TABLE ORDER_LINE69 ADD CONSTRAINT ORDER_LINE69CKC CHECK
(OL_W_ID BETWEEN 35361 AND 35880);
SET INTEGRITY FOR ORDER_LINE69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE70 OFF;
ALTER TABLE ORDER_LINE70 DROP CONSTRAINT ORDER_LINE70CKC;
ALTER TABLE ORDER_LINE70 ADD CONSTRAINT ORDER_LINE70CKC CHECK
(OL_W_ID BETWEEN 35881 AND 36400);
SET INTEGRITY FOR ORDER_LINE70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE71 OFF;
ALTER TABLE ORDER_LINE71 DROP CONSTRAINT ORDER_LINE71CKC;
ALTER TABLE ORDER_LINE71 ADD CONSTRAINT ORDER_LINE71CKC CHECK
(OL_W_ID BETWEEN 36401 AND 36920);
SET INTEGRITY FOR ORDER_LINE71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE72 OFF;
ALTER TABLE ORDER_LINE72 DROP CONSTRAINT ORDER_LINE72CKC;
ALTER TABLE ORDER_LINE72 ADD CONSTRAINT ORDER_LINE72CKC CHECK
(OL_W_ID BETWEEN 36921 AND 37440);
SET INTEGRITY FOR ORDER_LINE72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE73 OFF;
ALTER TABLE ORDER_LINE73 DROP CONSTRAINT ORDER_LINE73CKC;
ALTER TABLE ORDER_LINE73 ADD CONSTRAINT ORDER_LINE73CKC CHECK
(OL_W_ID BETWEEN 37441 AND 37960);
SET INTEGRITY FOR ORDER_LINE73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE74 OFF;
ALTER TABLE ORDER_LINE74 DROP CONSTRAINT ORDER_LINE74CKC;
ALTER TABLE ORDER_LINE74 ADD CONSTRAINT ORDER_LINE74CKC CHECK
(OL_W_ID BETWEEN 37961 AND 38480);
SET INTEGRITY FOR ORDER_LINE74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE75 OFF;
ALTER TABLE ORDER_LINE75 DROP CONSTRAINT ORDER_LINE75CKC;
ALTER TABLE ORDER_LINE75 ADD CONSTRAINT ORDER_LINE75CKC CHECK
(OL_W_ID BETWEEN 38481 AND 39000);
SET INTEGRITY FOR ORDER_LINE75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE76 OFF;
ALTER TABLE ORDER_LINE76 DROP CONSTRAINT ORDER_LINE76CKC;
ALTER TABLE ORDER_LINE76 ADD CONSTRAINT ORDER_LINE76CKC CHECK
(OL_W_ID BETWEEN 39001 AND 39520);
SET INTEGRITY FOR ORDER_LINE76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE77 OFF;
ALTER TABLE ORDER_LINE77 DROP CONSTRAINT ORDER_LINE77CKC;
ALTER TABLE ORDER_LINE77 ADD CONSTRAINT ORDER_LINE77CKC CHECK
(OL_W_ID BETWEEN 39521 AND 40040);
SET INTEGRITY FOR ORDER_LINE77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE78 OFF;
ALTER TABLE ORDER_LINE78 DROP CONSTRAINT ORDER_LINE78CKC;
ALTER TABLE ORDER_LINE78 ADD CONSTRAINT ORDER_LINE78CKC CHECK
(OL_W_ID BETWEEN 40041 AND 40560);
SET INTEGRITY FOR ORDER_LINE78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE79 OFF;
ALTER TABLE ORDER_LINE79 DROP CONSTRAINT ORDER_LINE79CKC;
ALTER TABLE ORDER_LINE79 ADD CONSTRAINT ORDER_LINE79CKC CHECK
(OL_W_ID BETWEEN 40561 AND 41080);
SET INTEGRITY FOR ORDER_LINE79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE80 OFF;
ALTER TABLE ORDER_LINE80 DROP CONSTRAINT ORDER_LINE80CKC;
ALTER TABLE ORDER_LINE80 ADD CONSTRAINT ORDER_LINE80CKC CHECK
(OL_W_ID >= 41081);
SET INTEGRITY FOR ORDER_LINE80 ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_STOCK.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK1 OFF;
ALTER TABLE STOCK1 DROP CONSTRAINT STOCK1CKC;
ALTER TABLE STOCK1 ADD CONSTRAINT STOCK1CKC CHECK (S_W_ID BETWEEN
1 AND 520);
SET INTEGRITY FOR STOCK1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK2 OFF;
ALTER TABLE STOCK2 DROP CONSTRAINT STOCK2CKC;
ALTER TABLE STOCK2 ADD CONSTRAINT STOCK2CKC CHECK (S_W_ID BETWEEN
521 AND 1040);
SET INTEGRITY FOR STOCK2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK3 OFF;
ALTER TABLE STOCK3 DROP CONSTRAINT STOCK3CKC;
ALTER TABLE STOCK3 ADD CONSTRAINT STOCK3CKC CHECK (S_W_ID BETWEEN
1041 AND 1560);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK4 OFF;
ALTER TABLE STOCK4 DROP CONSTRAINT STOCK4CKC;
ALTER TABLE STOCK4 ADD CONSTRAINT STOCK4CKC CHECK (S_W_ID BETWEEN
1561 AND 2080);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK5 OFF;
ALTER TABLE STOCK5 DROP CONSTRAINT STOCK5CKC;
ALTER TABLE STOCK5 ADD CONSTRAINT STOCK5CKC CHECK (S_W_ID BETWEEN
2081 AND 2600);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK6 OFF;
ALTER TABLE STOCK6 DROP CONSTRAINT STOCK6CKC;
ALTER TABLE STOCK6 ADD CONSTRAINT STOCK6CKC CHECK (S_W_ID BETWEEN
2601 AND 3120);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK7 OFF;
ALTER TABLE STOCK7 DROP CONSTRAINT STOCK7CKC;
ALTER TABLE STOCK7 ADD CONSTRAINT STOCK7CKC CHECK (S_W_ID BETWEEN
3121 AND 3640);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK8 OFF;
ALTER TABLE STOCK8 DROP CONSTRAINT STOCK8CKC;
ALTER TABLE STOCK8 ADD CONSTRAINT STOCK8CKC CHECK (S_W_ID BETWEEN
3641 AND 4160);
SET INTEGRITY FOR STOCK8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK9 OFF;
ALTER TABLE STOCK9 DROP CONSTRAINT STOCK9CKC;
ALTER TABLE STOCK9 ADD CONSTRAINT STOCK9CKC CHECK (S_W_ID BETWEEN
4161 AND 4680);
SET INTEGRITY FOR STOCK9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK10 OFF;
ALTER TABLE STOCK10 DROP CONSTRAINT STOCK10CKC;
ALTER TABLE STOCK10 ADD CONSTRAINT STOCK10CKC CHECK (S_W_ID
BETWEEN 4681 AND 5200);
SET INTEGRITY FOR STOCK10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK11 OFF;
ALTER TABLE STOCK11 DROP CONSTRAINT STOCK11CKC;
ALTER TABLE STOCK11 ADD CONSTRAINT STOCK11CKC CHECK (S_W_ID
BETWEEN 5201 AND 5720);
SET INTEGRITY FOR STOCK11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK12 OFF;
ALTER TABLE STOCK12 DROP CONSTRAINT STOCK12CKC;
```

```
ALTER TABLE STOCK12 ADD CONSTRAINT STOCK12CKC CHECK (S_W_ID
BETWEEN 5721 AND 6240);
SET INTEGRITY FOR STOCK12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK13 OFF;
ALTER TABLE STOCK13 DROP CONSTRAINT STOCK13CKC;
ALTER TABLE STOCK13 ADD CONSTRAINT STOCK13CKC CHECK (S_W_ID
BETWEEN 6241 AND 6760);
SET INTEGRITY FOR STOCK13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK14 OFF;
ALTER TABLE STOCK14 DROP CONSTRAINT STOCK14CKC;
ALTER TABLE STOCK14 ADD CONSTRAINT STOCK14CKC CHECK (S_W_ID
BETWEEN 6761 AND 7280);
SET INTEGRITY FOR STOCK14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK15 OFF;
ALTER TABLE STOCK15 DROP CONSTRAINT STOCK15CKC;
ALTER TABLE STOCK15 ADD CONSTRAINT STOCK15CKC CHECK (S_W_ID
BETWEEN 7281 AND 7800);
SET INTEGRITY FOR STOCK15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK16 OFF;
ALTER TABLE STOCK16 DROP CONSTRAINT STOCK16CKC;
ALTER TABLE STOCK16 ADD CONSTRAINT STOCK16CKC CHECK (S_W_ID
BETWEEN 7801 AND 8320);
SET INTEGRITY FOR STOCK16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK17 OFF;
ALTER TABLE STOCK17 DROP CONSTRAINT STOCK17CKC;
ALTER TABLE STOCK17 ADD CONSTRAINT STOCK17CKC CHECK (S_W_ID
BETWEEN 8321 AND 8840);
SET INTEGRITY FOR STOCK17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK18 OFF;
ALTER TABLE STOCK18 DROP CONSTRAINT STOCK18CKC;
ALTER TABLE STOCK18 ADD CONSTRAINT STOCK18CKC CHECK (S_W_ID
BETWEEN 8841 AND 9360);
SET INTEGRITY FOR STOCK18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK19 OFF;
ALTER TABLE STOCK19 DROP CONSTRAINT STOCK19CKC;
ALTER TABLE STOCK19 ADD CONSTRAINT STOCK19CKC CHECK (S_W_ID
BETWEEN 9361 AND 9880);
SET INTEGRITY FOR STOCK19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK20 OFF;
ALTER TABLE STOCK20 DROP CONSTRAINT STOCK20CKC;
ALTER TABLE STOCK20 ADD CONSTRAINT STOCK20CKC CHECK (S_W_ID
BETWEEN 9881 AND 10400);
SET INTEGRITY FOR STOCK20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK21 OFF;
ALTER TABLE STOCK21 DROP CONSTRAINT STOCK21CKC;
ALTER TABLE STOCK21 ADD CONSTRAINT STOCK21CKC CHECK (S_W_ID
BETWEEN 10401 AND 10920);
SET INTEGRITY FOR STOCK21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK22 OFF;
ALTER TABLE STOCK22 DROP CONSTRAINT STOCK22CKC;
ALTER TABLE STOCK22 ADD CONSTRAINT STOCK22CKC CHECK (S_W_ID
BETWEEN 10921 AND 11440);
SET INTEGRITY FOR STOCK22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK23 OFF;
ALTER TABLE STOCK23 DROP CONSTRAINT STOCK23CKC;
ALTER TABLE STOCK23 ADD CONSTRAINT STOCK23CKC CHECK (S_W_ID
BETWEEN 11441 AND 11960);
SET INTEGRITY FOR STOCK23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK24 OFF;
ALTER TABLE STOCK24 DROP CONSTRAINT STOCK24CKC;
ALTER TABLE STOCK24 ADD CONSTRAINT STOCK24CKC CHECK (S_W_ID
BETWEEN 11961 AND 12480);
SET INTEGRITY FOR STOCK24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK25 OFF;
ALTER TABLE STOCK25 DROP CONSTRAINT STOCK25CKC;
ALTER TABLE STOCK25 ADD CONSTRAINT STOCK25CKC CHECK (S_W_ID
BETWEEN 12481 AND 13000);
SET INTEGRITY FOR STOCK25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK26 OFF;
ALTER TABLE STOCK26 DROP CONSTRAINT STOCK26CKC;
ALTER TABLE STOCK26 ADD CONSTRAINT STOCK26CKC CHECK (S_W_ID
BETWEEN 13001 AND 13520);
SET INTEGRITY FOR STOCK26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK27 OFF;
ALTER TABLE STOCK27 DROP CONSTRAINT STOCK27CKC;
ALTER TABLE STOCK27 ADD CONSTRAINT STOCK27CKC CHECK (S_W_ID
BETWEEN 13521 AND 14040);
SET INTEGRITY FOR STOCK27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK28 OFF;
ALTER TABLE STOCK28 DROP CONSTRAINT STOCK28CKC;
ALTER TABLE STOCK28 ADD CONSTRAINT STOCK28CKC CHECK (S_W_ID
BETWEEN 14041 AND 14560);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK29 OFF;
ALTER TABLE STOCK29 DROP CONSTRAINT STOCK29CKC;
ALTER TABLE STOCK29 ADD CONSTRAINT STOCK29CKC CHECK (S_W_ID
BETWEEN 14561 AND 15080);
SET INTEGRITY FOR STOCK29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK30 OFF;
ALTER TABLE STOCK30 DROP CONSTRAINT STOCK30CKC;
ALTER TABLE STOCK30 ADD CONSTRAINT STOCK30CKC CHECK (S_W_ID
BETWEEN 15081 AND 15600);
SET INTEGRITY FOR STOCK30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK31 OFF;
ALTER TABLE STOCK31 DROP CONSTRAINT STOCK31CKC;
ALTER TABLE STOCK31 ADD CONSTRAINT STOCK31CKC CHECK (S_W_ID
BETWEEN 15601 AND 16120);
SET INTEGRITY FOR STOCK31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK32 OFF;
ALTER TABLE STOCK32 DROP CONSTRAINT STOCK32CKC;
ALTER TABLE STOCK32 ADD CONSTRAINT STOCK32CKC CHECK (S_W_ID
BETWEEN 16121 AND 16640);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK33 OFF;
ALTER TABLE STOCK33 DROP CONSTRAINT STOCK33CKC;
ALTER TABLE STOCK33 ADD CONSTRAINT STOCK33CKC CHECK (S_W_ID
BETWEEN 16641 AND 17160);
SET INTEGRITY FOR STOCK33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK34 OFF;
ALTER TABLE STOCK34 DROP CONSTRAINT STOCK34CKC;
ALTER TABLE STOCK34 ADD CONSTRAINT STOCK34CKC CHECK (S_W_ID
BETWEEN 17161 AND 17680);
SET INTEGRITY FOR STOCK34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK35 OFF;
ALTER TABLE STOCK35 DROP CONSTRAINT STOCK35CKC;
ALTER TABLE STOCK35 ADD CONSTRAINT STOCK35CKC CHECK (S_W_ID
BETWEEN 17681 AND 18200);
SET INTEGRITY FOR STOCK35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK36 OFF;
ALTER TABLE STOCK36 DROP CONSTRAINT STOCK36CKC;
ALTER TABLE STOCK36 ADD CONSTRAINT STOCK36CKC CHECK (S_W_ID
BETWEEN 18201 AND 18720);
SET INTEGRITY FOR STOCK36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK37 OFF;
ALTER TABLE STOCK37 DROP CONSTRAINT STOCK37CKC;
ALTER TABLE STOCK37 ADD CONSTRAINT STOCK37CKC CHECK (S_W_ID
BETWEEN 18721 AND 19240);
SET INTEGRITY FOR STOCK37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK38 OFF;
ALTER TABLE STOCK38 DROP CONSTRAINT STOCK38CKC;
ALTER TABLE STOCK38 ADD CONSTRAINT STOCK38CKC CHECK (S_W_ID
BETWEEN 19241 AND 19760);
SET INTEGRITY FOR STOCK38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK39 OFF;
ALTER TABLE STOCK39 DROP CONSTRAINT STOCK39CKC;
ALTER TABLE STOCK39 ADD CONSTRAINT STOCK39CKC CHECK (S_W_ID
BETWEEN 19761 AND 20280);
SET INTEGRITY FOR STOCK39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK40 OFF;
ALTER TABLE STOCK40 DROP CONSTRAINT STOCK40CKC;
ALTER TABLE STOCK40 ADD CONSTRAINT STOCK40CKC CHECK (S_W_ID
BETWEEN 20281 AND 20800);
SET INTEGRITY FOR STOCK40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK41 OFF;
ALTER TABLE STOCK41 DROP CONSTRAINT STOCK41CKC;
ALTER TABLE STOCK41 ADD CONSTRAINT STOCK41CKC CHECK (S_W_ID
BETWEEN 20801 AND 21320);
SET INTEGRITY FOR STOCK41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK42 OFF;
ALTER TABLE STOCK42 DROP CONSTRAINT STOCK42CKC;
```

```
ALTER TABLE STOCK42 ADD CONSTRAINT STOCK42CKC CHECK (S_W_ID
BETWEEN 21321 AND 21840);
SET INTEGRITY FOR STOCK42 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK43 OFF;
ALTER TABLE STOCK43 DROP CONSTRAINT STOCK43CKC;
ALTER TABLE STOCK43 ADD CONSTRAINT STOCK43CKC CHECK (S_W_ID
BETWEEN 21841 AND 22360);
SET INTEGRITY FOR STOCK43 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK44 OFF;
ALTER TABLE STOCK44 DROP CONSTRAINT STOCK44CKC;
ALTER TABLE STOCK44 ADD CONSTRAINT STOCK44CKC CHECK (S_W_ID
BETWEEN 22361 AND 22880);
SET INTEGRITY FOR STOCK44 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK45 OFF;
ALTER TABLE STOCK45 DROP CONSTRAINT STOCK45CKC;
ALTER TABLE STOCK45 ADD CONSTRAINT STOCK45CKC CHECK (S_W_ID
BETWEEN 22881 AND 23400);
SET INTEGRITY FOR STOCK45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK46 OFF;
ALTER TABLE STOCK46 DROP CONSTRAINT STOCK46CKC;
ALTER TABLE STOCK46 ADD CONSTRAINT STOCK46CKC CHECK (S_W_ID
BETWEEN 23401 AND 23920);
SET INTEGRITY FOR STOCK46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK47 OFF;
ALTER TABLE STOCK47 DROP CONSTRAINT STOCK47CKC;
ALTER TABLE STOCK47 ADD CONSTRAINT STOCK47CKC CHECK (S_W_ID
BETWEEN 23921 AND 24440);
SET INTEGRITY FOR STOCK47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK48 OFF;
ALTER TABLE STOCK48 DROP CONSTRAINT STOCK48CKC;
ALTER TABLE STOCK48 ADD CONSTRAINT STOCK48CKC CHECK (S_W_ID
BETWEEN 24441 AND 24960);
SET INTEGRITY FOR STOCK48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK49 OFF;
ALTER TABLE STOCK49 DROP CONSTRAINT STOCK49CKC;
ALTER TABLE STOCK49 ADD CONSTRAINT STOCK49CKC CHECK (S_W_ID
BETWEEN 24961 AND 25480);
SET INTEGRITY FOR STOCK49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK50 OFF;
ALTER TABLE STOCK50 DROP CONSTRAINT STOCK50CKC;
ALTER TABLE STOCK50 ADD CONSTRAINT STOCK50CKC CHECK (S_W_ID
BETWEEN 25481 AND 26000);
SET INTEGRITY FOR STOCK50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK51 OFF;
ALTER TABLE STOCK51 DROP CONSTRAINT STOCK51CKC;
ALTER TABLE STOCK51 ADD CONSTRAINT STOCK51CKC CHECK (S_W_ID
BETWEEN 26001 AND 26520);
SET INTEGRITY FOR STOCK51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK52 OFF;
ALTER TABLE STOCK52 DROP CONSTRAINT STOCK52CKC;
ALTER TABLE STOCK52 ADD CONSTRAINT STOCK52CKC CHECK (S_W_ID
BETWEEN 26521 AND 27040);
SET INTEGRITY FOR STOCK52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK53 OFF;
ALTER TABLE STOCK53 DROP CONSTRAINT STOCK53CKC;
ALTER TABLE STOCK53 ADD CONSTRAINT STOCK53CKC CHECK (S_W_ID
BETWEEN 27041 AND 27560);
SET INTEGRITY FOR STOCK53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK54 OFF;
ALTER TABLE STOCK54 DROP CONSTRAINT STOCK54CKC;
ALTER TABLE STOCK54 ADD CONSTRAINT STOCK54CKC CHECK (S_W_ID
BETWEEN 27561 AND 28080);
SET INTEGRITY FOR STOCK54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK55 OFF;
ALTER TABLE STOCK55 DROP CONSTRAINT STOCK55CKC;
ALTER TABLE STOCK55 ADD CONSTRAINT STOCK55CKC CHECK (S_W_ID
BETWEEN 28081 AND 28600);
SET INTEGRITY FOR STOCK55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK56 OFF;
ALTER TABLE STOCK56 DROP CONSTRAINT STOCK56CKC;
ALTER TABLE STOCK56 ADD CONSTRAINT STOCK56CKC CHECK (S_W_ID
BETWEEN 28601 AND 29120);
SET INTEGRITY FOR STOCK56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK57 OFF;
ALTER TABLE STOCK57 DROP CONSTRAINT STOCK57CKC;
ALTER TABLE STOCK57 ADD CONSTRAINT STOCK57CKC CHECK (S_W_ID
BETWEEN 29121 AND 29640);
SET INTEGRITY FOR STOCK57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK58 OFF;
ALTER TABLE STOCK58 DROP CONSTRAINT STOCK58CKC;
ALTER TABLE STOCK58 ADD CONSTRAINT STOCK58CKC CHECK (S_W_ID
BETWEEN 29641 AND 30160);
SET INTEGRITY FOR STOCK58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK59 OFF;
ALTER TABLE STOCK59 DROP CONSTRAINT STOCK59CKC;
ALTER TABLE STOCK59 ADD CONSTRAINT STOCK59CKC CHECK (S_W_ID
BETWEEN 30161 AND 30680);
SET INTEGRITY FOR STOCK59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK60 OFF;
ALTER TABLE STOCK60 DROP CONSTRAINT STOCK60CKC;
ALTER TABLE STOCK60 ADD CONSTRAINT STOCK60CKC CHECK (S_W_ID
BETWEEN 30681 AND 31200);
SET INTEGRITY FOR STOCK60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK61 OFF;
ALTER TABLE STOCK61 DROP CONSTRAINT STOCK61CKC;
ALTER TABLE STOCK61 ADD CONSTRAINT STOCK61CKC CHECK (S_W_ID
BETWEEN 31201 AND 31720);
SET INTEGRITY FOR STOCK61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK62 OFF;
ALTER TABLE STOCK62 DROP CONSTRAINT STOCK62CKC;
ALTER TABLE STOCK62 ADD CONSTRAINT STOCK62CKC CHECK (S_W_ID
BETWEEN 31721 AND 32240);
SET INTEGRITY FOR STOCK62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK63 OFF;
ALTER TABLE STOCK63 DROP CONSTRAINT STOCK63CKC;
ALTER TABLE STOCK63 ADD CONSTRAINT STOCK63CKC CHECK (S_W_ID
BETWEEN 32241 AND 32760);
SET INTEGRITY FOR STOCK63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK64 OFF;
ALTER TABLE STOCK64 DROP CONSTRAINT STOCK64CKC;
ALTER TABLE STOCK64 ADD CONSTRAINT STOCK64CKC CHECK (S_W_ID
BETWEEN 32761 AND 33280);
SET INTEGRITY FOR STOCK64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK65 OFF;
ALTER TABLE STOCK65 DROP CONSTRAINT STOCK65CKC;
ALTER TABLE STOCK65 ADD CONSTRAINT STOCK65CKC CHECK (S_W_ID
BETWEEN 33281 AND 33800);
SET INTEGRITY FOR STOCK65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK66 OFF;
ALTER TABLE STOCK66 DROP CONSTRAINT STOCK66CKC;
ALTER TABLE STOCK66 ADD CONSTRAINT STOCK66CKC CHECK (S_W_ID
BETWEEN 33801 AND 34320);
SET INTEGRITY FOR STOCK66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK67 OFF;
ALTER TABLE STOCK67 DROP CONSTRAINT STOCK67CKC;
ALTER TABLE STOCK67 ADD CONSTRAINT STOCK67CKC CHECK (S_W_ID
BETWEEN 34321 AND 34840);
SET INTEGRITY FOR STOCK67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK68 OFF;
ALTER TABLE STOCK68 DROP CONSTRAINT STOCK68CKC;
ALTER TABLE STOCK68 ADD CONSTRAINT STOCK68CKC CHECK (S_W_ID
BETWEEN 34841 AND 35360);
SET INTEGRITY FOR STOCK68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK69 OFF;
ALTER TABLE STOCK69 DROP CONSTRAINT STOCK69CKC;
ALTER TABLE STOCK69 ADD CONSTRAINT STOCK69CKC CHECK (S_W_ID
BETWEEN 35361 AND 35880);
SET INTEGRITY FOR STOCK69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK70 OFF;
ALTER TABLE STOCK70 DROP CONSTRAINT STOCK70CKC;
ALTER TABLE STOCK70 ADD CONSTRAINT STOCK70CKC CHECK (S_W_ID
BETWEEN 35881 AND 36400);
SET INTEGRITY FOR STOCK70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK71 OFF;
ALTER TABLE STOCK71 DROP CONSTRAINT STOCK71CKC;
ALTER TABLE STOCK71 ADD CONSTRAINT STOCK71CKC CHECK (S_W_ID
BETWEEN 36401 AND 36920);
SET INTEGRITY FOR STOCK71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK72 OFF;
ALTER TABLE STOCK72 DROP CONSTRAINT STOCK72CKC;
```

```
ALTER TABLE STOCK72 ADD CONSTRAINT STOCK72CKC CHECK (S_W_ID
BETWEEN 36921 AND 37440);
SET INTEGRITY FOR STOCK72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK73 OFF;
ALTER TABLE STOCK73 DROP CONSTRAINT STOCK73CKC;
ALTER TABLE STOCK73 ADD CONSTRAINT STOCK73CKC CHECK (S_W_ID
BETWEEN 37441 AND 37960);
SET INTEGRITY FOR STOCK73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK74 OFF;
ALTER TABLE STOCK74 DROP CONSTRAINT STOCK74CKC;
ALTER TABLE STOCK74 ADD CONSTRAINT STOCK74CKC CHECK (S_W_ID
BETWEEN 37961 AND 38480);
SET INTEGRITY FOR STOCK74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK75 OFF;
ALTER TABLE STOCK75 DROP CONSTRAINT STOCK75CKC;
ALTER TABLE STOCK75 ADD CONSTRAINT STOCK75CKC CHECK (S_W_ID
BETWEEN 38481 AND 39000);
SET INTEGRITY FOR STOCK75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK76 OFF;
ALTER TABLE STOCK76 DROP CONSTRAINT STOCK76CKC;
ALTER TABLE STOCK76 ADD CONSTRAINT STOCK76CKC CHECK (S_W_ID
BETWEEN 39001 AND 39520);
SET INTEGRITY FOR STOCK76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK77 OFF;
ALTER TABLE STOCK77 DROP CONSTRAINT STOCK77CKC;
ALTER TABLE STOCK77 ADD CONSTRAINT STOCK77CKC CHECK (S_W_ID
BETWEEN 39521 AND 40040);
SET INTEGRITY FOR STOCK77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK78 OFF;
ALTER TABLE STOCK78 DROP CONSTRAINT STOCK78CKC;
ALTER TABLE STOCK78 ADD CONSTRAINT STOCK78CKC CHECK (S_W_ID
BETWEEN 40041 AND 40560);
SET INTEGRITY FOR STOCK78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK79 OFF;
ALTER TABLE STOCK79 DROP CONSTRAINT STOCK79CKC;
ALTER TABLE STOCK79 ADD CONSTRAINT STOCK79CKC CHECK (S_W_ID
BETWEEN 40561 AND 41080);
SET INTEGRITY FOR STOCK79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK80 OFF;
ALTER TABLE STOCK80 DROP CONSTRAINT STOCK80CKC;
ALTER TABLE STOCK80 ADD CONSTRAINT STOCK80CKC CHECK (S_W_ID >=
41081);
SET INTEGRITY FOR STOCK80 ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_WAREHOUSE.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE1 OFF;
ALTER TABLE WAREHOUSE1 DROP CONSTRAINT WAREHOUSE1CKC;
ALTER TABLE WAREHOUSE1 ADD CONSTRAINT WAREHOUSE1CKC CHECK (W_ID
BETWEEN 1 AND 520);
SET INTEGRITY FOR WAREHOUSE1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE2 OFF;
ALTER TABLE WAREHOUSE2 DROP CONSTRAINT WAREHOUSE2CKC;
ALTER TABLE WAREHOUSE2 ADD CONSTRAINT WAREHOUSE2CKC CHECK (W_ID
BETWEEN 521 AND 1040);
SET INTEGRITY FOR WAREHOUSE2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE3 OFF;
ALTER TABLE WAREHOUSE3 DROP CONSTRAINT WAREHOUSE3CKC;
ALTER TABLE WAREHOUSE3 ADD CONSTRAINT WAREHOUSE3CKC CHECK (W_ID
BETWEEN 1041 AND 1560);
SET INTEGRITY FOR WAREHOUSE3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE4 OFF;
ALTER TABLE WAREHOUSE4 DROP CONSTRAINT WAREHOUSE4CKC;
ALTER TABLE WAREHOUSE4 ADD CONSTRAINT WAREHOUSE4CKC CHECK (W_ID
BETWEEN 1561 AND 2080);
SET INTEGRITY FOR WAREHOUSE4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE5 OFF;
ALTER TABLE WAREHOUSE5 DROP CONSTRAINT WAREHOUSE5CKC;
ALTER TABLE WAREHOUSE5 ADD CONSTRAINT WAREHOUSE5CKC CHECK (W_ID
BETWEEN 2081 AND 2600);
SET INTEGRITY FOR WAREHOUSE5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE6 OFF;
ALTER TABLE WAREHOUSE6 DROP CONSTRAINT WAREHOUSE6CKC;
ALTER TABLE WAREHOUSE6 ADD CONSTRAINT WAREHOUSE6CKC CHECK (W_ID
BETWEEN 2601 AND 3120);
SET INTEGRITY FOR WAREHOUSE6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE7 OFF;
ALTER TABLE WAREHOUSE7 DROP CONSTRAINT WAREHOUSE7CKC;
ALTER TABLE WAREHOUSE7 ADD CONSTRAINT WAREHOUSE7CKC CHECK (W_ID
BETWEEN 3121 AND 3640);
SET INTEGRITY FOR WAREHOUSE7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE8 OFF;
ALTER TABLE WAREHOUSE8 DROP CONSTRAINT WAREHOUSE8CKC;
ALTER TABLE WAREHOUSE8 ADD CONSTRAINT WAREHOUSE8CKC CHECK (W_ID
BETWEEN 3641 AND 4160);
SET INTEGRITY FOR WAREHOUSE8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE9 OFF;
ALTER TABLE WAREHOUSE9 DROP CONSTRAINT WAREHOUSE9CKC;
ALTER TABLE WAREHOUSE9 ADD CONSTRAINT WAREHOUSE9CKC CHECK (W_ID
BETWEEN 4161 AND 4680);
SET INTEGRITY FOR WAREHOUSE9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE10 OFF;
ALTER TABLE WAREHOUSE10 DROP CONSTRAINT WAREHOUSE10CKC;
ALTER TABLE WAREHOUSE10 ADD CONSTRAINT WAREHOUSE10CKC CHECK
(W_ID BETWEEN 4681 AND 5200);
SET INTEGRITY FOR WAREHOUSE10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE11 OFF;
ALTER TABLE WAREHOUSE11 DROP CONSTRAINT WAREHOUSE11CKC;
ALTER TABLE WAREHOUSE11 ADD CONSTRAINT WAREHOUSE11CKC CHECK
(W_ID BETWEEN 5201 AND 5720);
SET INTEGRITY FOR WAREHOUSE11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE12 OFF;
ALTER TABLE WAREHOUSE12 DROP CONSTRAINT WAREHOUSE12CKC;
ALTER TABLE WAREHOUSE12 ADD CONSTRAINT WAREHOUSE12CKC CHECK
(W_ID BETWEEN 5721 AND 6240);
SET INTEGRITY FOR WAREHOUSE12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE13 OFF;
ALTER TABLE WAREHOUSE13 DROP CONSTRAINT WAREHOUSE13CKC;
ALTER TABLE WAREHOUSE13 ADD CONSTRAINT WAREHOUSE13CKC CHECK
(W_ID BETWEEN 6241 AND 6760);
SET INTEGRITY FOR WAREHOUSE13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE14 OFF;
ALTER TABLE WAREHOUSE14 DROP CONSTRAINT WAREHOUSE14CKC;
ALTER TABLE WAREHOUSE14 ADD CONSTRAINT WAREHOUSE14CKC CHECK
(W_ID BETWEEN 6761 AND 7280);
SET INTEGRITY FOR WAREHOUSE14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE15 OFF;
ALTER TABLE WAREHOUSE15 DROP CONSTRAINT WAREHOUSE15CKC;
ALTER TABLE WAREHOUSE15 ADD CONSTRAINT WAREHOUSE15CKC CHECK
(W_ID BETWEEN 7281 AND 7800);
SET INTEGRITY FOR WAREHOUSE15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE16 OFF;
ALTER TABLE WAREHOUSE16 DROP CONSTRAINT WAREHOUSE16CKC;
ALTER TABLE WAREHOUSE16 ADD CONSTRAINT WAREHOUSE16CKC CHECK
(W_ID BETWEEN 7801 AND 8320);
SET INTEGRITY FOR WAREHOUSE16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE17 OFF;
ALTER TABLE WAREHOUSE17 DROP CONSTRAINT WAREHOUSE17CKC;
ALTER TABLE WAREHOUSE17 ADD CONSTRAINT WAREHOUSE17CKC CHECK
(W_ID BETWEEN 8321 AND 8840);
SET INTEGRITY FOR WAREHOUSE17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE18 OFF;
ALTER TABLE WAREHOUSE18 DROP CONSTRAINT WAREHOUSE18CKC;
ALTER TABLE WAREHOUSE18 ADD CONSTRAINT WAREHOUSE18CKC CHECK
(W_ID BETWEEN 8841 AND 9360);
SET INTEGRITY FOR WAREHOUSE18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE19 OFF;
ALTER TABLE WAREHOUSE19 DROP CONSTRAINT WAREHOUSE19CKC;
ALTER TABLE WAREHOUSE19 ADD CONSTRAINT WAREHOUSE19CKC CHECK
(W_ID BETWEEN 9361 AND 9880);
SET INTEGRITY FOR WAREHOUSE19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE20 OFF;
ALTER TABLE WAREHOUSE20 DROP CONSTRAINT WAREHOUSE20CKC;
ALTER TABLE WAREHOUSE20 ADD CONSTRAINT WAREHOUSE20CKC CHECK
(W_ID BETWEEN 9881 AND 10400);
SET INTEGRITY FOR WAREHOUSE20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE21 OFF;
ALTER TABLE WAREHOUSE21 DROP CONSTRAINT WAREHOUSE21CKC;
ALTER TABLE WAREHOUSE21 ADD CONSTRAINT WAREHOUSE21CKC CHECK
(W_ID BETWEEN 10401 AND 10920);
SET INTEGRITY FOR WAREHOUSE21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
```

```
SET INTEGRITY FOR WAREHOUSE22 OFF;
ALTER TABLE WAREHOUSE22 DROP CONSTRAINT WAREHOUSE22CKC;
ALTER TABLE WAREHOUSE22 ADD CONSTRAINT WAREHOUSE22CKC CHECK
(W_ID BETWEEN 10921 AND 11440);
SET INTEGRITY FOR WAREHOUSE22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE23 OFF;
ALTER TABLE WAREHOUSE23 DROP CONSTRAINT WAREHOUSE23CKC;
ALTER TABLE WAREHOUSE23 ADD CONSTRAINT WAREHOUSE23CKC CHECK
(W_ID BETWEEN 11441 AND 11960);
SET INTEGRITY FOR WAREHOUSE23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE24 OFF;
ALTER TABLE WAREHOUSE24 DROP CONSTRAINT WAREHOUSE24CKC;
ALTER TABLE WAREHOUSE24 ADD CONSTRAINT WAREHOUSE24CKC CHECK
(W_ID BETWEEN 11961 AND 12480);
SET INTEGRITY FOR WAREHOUSE24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE25 OFF;
ALTER TABLE WAREHOUSE25 DROP CONSTRAINT WAREHOUSE25CKC;
ALTER TABLE WAREHOUSE25 ADD CONSTRAINT WAREHOUSE25CKC CHECK
(W_ID BETWEEN 12481 AND 13000);
SET INTEGRITY FOR WAREHOUSE25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE26 OFF;
ALTER TABLE WAREHOUSE26 DROP CONSTRAINT WAREHOUSE26CKC;
ALTER TABLE WAREHOUSE26 ADD CONSTRAINT WAREHOUSE26CKC CHECK
(W_ID BETWEEN 13001 AND 13520);
SET INTEGRITY FOR WAREHOUSE26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE27 OFF;
ALTER TABLE WAREHOUSE27 DROP CONSTRAINT WAREHOUSE27CKC;
ALTER TABLE WAREHOUSE27 ADD CONSTRAINT WAREHOUSE27CKC CHECK
(W_ID BETWEEN 13521 AND 14040);
SET INTEGRITY FOR WAREHOUSE27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE28 OFF;
ALTER TABLE WAREHOUSE28 DROP CONSTRAINT WAREHOUSE28CKC;
ALTER TABLE WAREHOUSE28 ADD CONSTRAINT WAREHOUSE28CKC CHECK
(W_ID BETWEEN 14041 AND 14560);
SET INTEGRITY FOR WAREHOUSE28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE29 OFF;
ALTER TABLE WAREHOUSE29 DROP CONSTRAINT WAREHOUSE29CKC;
ALTER TABLE WAREHOUSE29 ADD CONSTRAINT WAREHOUSE29CKC CHECK
(W_ID BETWEEN 14561 AND 15080);
SET INTEGRITY FOR WAREHOUSE29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE30 OFF;
ALTER TABLE WAREHOUSE30 DROP CONSTRAINT WAREHOUSE30CKC;
ALTER TABLE WAREHOUSE30 ADD CONSTRAINT WAREHOUSE30CKC CHECK
(W_ID BETWEEN 15081 AND 15600);
SET INTEGRITY FOR WAREHOUSE30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE31 OFF;
ALTER TABLE WAREHOUSE31 DROP CONSTRAINT WAREHOUSE31CKC;
ALTER TABLE WAREHOUSE31 ADD CONSTRAINT WAREHOUSE31CKC CHECK
(W_ID BETWEEN 15601 AND 16120);
SET INTEGRITY FOR WAREHOUSE31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE32 OFF;
ALTER TABLE WAREHOUSE32 DROP CONSTRAINT WAREHOUSE32CKC;
ALTER TABLE WAREHOUSE32 ADD CONSTRAINT WAREHOUSE32CKC CHECK
(W_ID BETWEEN 16121 AND 16640);
SET INTEGRITY FOR WAREHOUSE32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE33 OFF;
ALTER TABLE WAREHOUSE33 DROP CONSTRAINT WAREHOUSE33CKC;
ALTER TABLE WAREHOUSE33 ADD CONSTRAINT WAREHOUSE33CKC CHECK
(W_ID BETWEEN 16641 AND 17160);
SET INTEGRITY FOR WAREHOUSE33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE34 OFF;
ALTER TABLE WAREHOUSE34 DROP CONSTRAINT WAREHOUSE34CKC;
ALTER TABLE WAREHOUSE34 ADD CONSTRAINT WAREHOUSE34CKC CHECK
(W_ID BETWEEN 17161 AND 17680);
SET INTEGRITY FOR WAREHOUSE34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE35 OFF;
ALTER TABLE WAREHOUSE35 DROP CONSTRAINT WAREHOUSE35CKC;
ALTER TABLE WAREHOUSE35 ADD CONSTRAINT WAREHOUSE35CKC CHECK
(W_ID BETWEEN 17681 AND 18200);
SET INTEGRITY FOR WAREHOUSE35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE36 OFF;
ALTER TABLE WAREHOUSE36 DROP CONSTRAINT WAREHOUSE36CKC;
ALTER TABLE WAREHOUSE36 ADD CONSTRAINT WAREHOUSE36CKC CHECK
(W_ID BETWEEN 18201 AND 18720);
SET INTEGRITY FOR WAREHOUSE36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE37 OFF;
ALTER TABLE WAREHOUSE37 DROP CONSTRAINT WAREHOUSE37CKC;
ALTER TABLE WAREHOUSE37 ADD CONSTRAINT WAREHOUSE37CKC CHECK
(W_ID BETWEEN 18721 AND 19240);
SET INTEGRITY FOR WAREHOUSE37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE38 OFF;
ALTER TABLE WAREHOUSE38 DROP CONSTRAINT WAREHOUSE38CKC;
ALTER TABLE WAREHOUSE38 ADD CONSTRAINT WAREHOUSE38CKC CHECK
(W_ID BETWEEN 19241 AND 19760);
SET INTEGRITY FOR WAREHOUSE38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE39 OFF;
ALTER TABLE WAREHOUSE39 DROP CONSTRAINT WAREHOUSE39CKC;
ALTER TABLE WAREHOUSE39 ADD CONSTRAINT WAREHOUSE39CKC CHECK
(W_ID BETWEEN 19761 AND 20280);
SET INTEGRITY FOR WAREHOUSE39 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE40 OFF;
ALTER TABLE WAREHOUSE40 DROP CONSTRAINT WAREHOUSE40CKC;
ALTER TABLE WAREHOUSE40 ADD CONSTRAINT WAREHOUSE40CKC CHECK
(W_ID BETWEEN 20281 AND 20800);
SET INTEGRITY FOR WAREHOUSE40 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE41 OFF;
ALTER TABLE WAREHOUSE41 DROP CONSTRAINT WAREHOUSE41CKC;
ALTER TABLE WAREHOUSE41 ADD CONSTRAINT WAREHOUSE41CKC CHECK
(W_ID BETWEEN 20801 AND 21320);
SET INTEGRITY FOR WAREHOUSE41 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE42 OFF;
ALTER TABLE WAREHOUSE42 DROP CONSTRAINT WAREHOUSE42CKC;
ALTER TABLE WAREHOUSE42 ADD CONSTRAINT WAREHOUSE42CKC CHECK
(W_ID BETWEEN 21321 AND 21840);
SET INTEGRITY FOR WAREHOUSE42 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE43 OFF;
ALTER TABLE WAREHOUSE43 DROP CONSTRAINT WAREHOUSE43CKC;
ALTER TABLE WAREHOUSE43 ADD CONSTRAINT WAREHOUSE43CKC CHECK
(W_ID BETWEEN 21841 AND 22360);
SET INTEGRITY FOR WAREHOUSE43 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE44 OFF;
ALTER TABLE WAREHOUSE44 DROP CONSTRAINT WAREHOUSE44CKC;
ALTER TABLE WAREHOUSE44 ADD CONSTRAINT WAREHOUSE44CKC CHECK
(W_ID BETWEEN 22361 AND 22880);
SET INTEGRITY FOR WAREHOUSE44 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE45 OFF;
ALTER TABLE WAREHOUSE45 DROP CONSTRAINT WAREHOUSE45CKC;
ALTER TABLE WAREHOUSE45 ADD CONSTRAINT WAREHOUSE45CKC CHECK
(W_ID BETWEEN 22881 AND 23400);
SET INTEGRITY FOR WAREHOUSE45 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE46 OFF;
ALTER TABLE WAREHOUSE46 DROP CONSTRAINT WAREHOUSE46CKC;
ALTER TABLE WAREHOUSE46 ADD CONSTRAINT WAREHOUSE46CKC CHECK
(W_ID BETWEEN 23401 AND 23920);
SET INTEGRITY FOR WAREHOUSE46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE47 OFF;
ALTER TABLE WAREHOUSE47 DROP CONSTRAINT WAREHOUSE47CKC;
ALTER TABLE WAREHOUSE47 ADD CONSTRAINT WAREHOUSE47CKC CHECK
(W_ID BETWEEN 23921 AND 24440);
SET INTEGRITY FOR WAREHOUSE47 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE48 OFF;
ALTER TABLE WAREHOUSE48 DROP CONSTRAINT WAREHOUSE48CKC;
ALTER TABLE WAREHOUSE48 ADD CONSTRAINT WAREHOUSE48CKC CHECK
(W_ID BETWEEN 24441 AND 24960);
SET INTEGRITY FOR WAREHOUSE48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE49 OFF;
ALTER TABLE WAREHOUSE49 DROP CONSTRAINT WAREHOUSE49CKC;
ALTER TABLE WAREHOUSE49 ADD CONSTRAINT WAREHOUSE49CKC CHECK
(W_ID BETWEEN 24961 AND 25480);
SET INTEGRITY FOR WAREHOUSE49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE50 OFF;
ALTER TABLE WAREHOUSE50 DROP CONSTRAINT WAREHOUSE50CKC;
ALTER TABLE WAREHOUSE50 ADD CONSTRAINT WAREHOUSE50CKC CHECK
(W_ID BETWEEN 25481 AND 26000);
SET INTEGRITY FOR WAREHOUSE50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE51 OFF;
ALTER TABLE WAREHOUSE51 DROP CONSTRAINT WAREHOUSE51CKC;
ALTER TABLE WAREHOUSE51 ADD CONSTRAINT WAREHOUSE51CKC CHECK
(W_ID BETWEEN 26001 AND 26520);
SET INTEGRITY FOR WAREHOUSE51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
```

```
SET INTEGRITY FOR WAREHOUSE52 OFF;
ALTER TABLE WAREHOUSE52 DROP CONSTRAINT WAREHOUSE52CKC;
ALTER TABLE WAREHOUSE52 ADD CONSTRAINT WAREHOUSE52CKC CHECK
(W_ID BETWEEN 26521 AND 27040);
SET INTEGRITY FOR WAREHOUSE52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE53 OFF;
ALTER TABLE WAREHOUSE53 DROP CONSTRAINT WAREHOUSE53CKC;
ALTER TABLE WAREHOUSE53 ADD CONSTRAINT WAREHOUSE53CKC CHECK
(W_ID BETWEEN 27041 AND 27560);
SET INTEGRITY FOR WAREHOUSE53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE54 OFF;
ALTER TABLE WAREHOUSE54 DROP CONSTRAINT WAREHOUSE54CKC;
ALTER TABLE WAREHOUSE54 ADD CONSTRAINT WAREHOUSE54CKC CHECK
(W_ID BETWEEN 27561 AND 28080);
SET INTEGRITY FOR WAREHOUSE54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE55 OFF;
ALTER TABLE WAREHOUSE55 DROP CONSTRAINT WAREHOUSE55CKC;
ALTER TABLE WAREHOUSE55 ADD CONSTRAINT WAREHOUSE55CKC CHECK
(W_ID BETWEEN 28081 AND 28600);
SET INTEGRITY FOR WAREHOUSE55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE56 OFF;
ALTER TABLE WAREHOUSE56 DROP CONSTRAINT WAREHOUSE56CKC;
ALTER TABLE WAREHOUSE56 ADD CONSTRAINT WAREHOUSE56CKC CHECK
(W_ID BETWEEN 28601 AND 29120);
SET INTEGRITY FOR WAREHOUSE56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE57 OFF;
ALTER TABLE WAREHOUSE57 DROP CONSTRAINT WAREHOUSE57CKC;
ALTER TABLE WAREHOUSE57 ADD CONSTRAINT WAREHOUSE57CKC CHECK
(W_ID BETWEEN 29121 AND 29640);
SET INTEGRITY FOR WAREHOUSE57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE58 OFF;
ALTER TABLE WAREHOUSE58 DROP CONSTRAINT WAREHOUSE58CKC;
ALTER TABLE WAREHOUSE58 ADD CONSTRAINT WAREHOUSE58CKC CHECK
(W_ID BETWEEN 29641 AND 30160);
SET INTEGRITY FOR WAREHOUSE58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE59 OFF;
ALTER TABLE WAREHOUSE59 DROP CONSTRAINT WAREHOUSE59CKC;
ALTER TABLE WAREHOUSE59 ADD CONSTRAINT WAREHOUSE59CKC CHECK
(W_ID BETWEEN 30161 AND 30680);
SET INTEGRITY FOR WAREHOUSE59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE60 OFF;
ALTER TABLE WAREHOUSE60 DROP CONSTRAINT WAREHOUSE60CKC;
ALTER TABLE WAREHOUSE60 ADD CONSTRAINT WAREHOUSE60CKC CHECK
(W_ID BETWEEN 30681 AND 31200);
SET INTEGRITY FOR WAREHOUSE60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE61 OFF;
ALTER TABLE WAREHOUSE61 DROP CONSTRAINT WAREHOUSE61CKC;
ALTER TABLE WAREHOUSE61 ADD CONSTRAINT WAREHOUSE61CKC CHECK
(W_ID BETWEEN 31201 AND 31720);
SET INTEGRITY FOR WAREHOUSE61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE62 OFF;
ALTER TABLE WAREHOUSE62 DROP CONSTRAINT WAREHOUSE62CKC;
ALTER TABLE WAREHOUSE62 ADD CONSTRAINT WAREHOUSE62CKC CHECK
(W_ID BETWEEN 31721 AND 32240);
SET INTEGRITY FOR WAREHOUSE62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE63 OFF;
ALTER TABLE WAREHOUSE63 DROP CONSTRAINT WAREHOUSE63CKC;
ALTER TABLE WAREHOUSE63 ADD CONSTRAINT WAREHOUSE63CKC CHECK
(W_ID BETWEEN 32241 AND 32760);
SET INTEGRITY FOR WAREHOUSE63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE64 OFF;
ALTER TABLE WAREHOUSE64 DROP CONSTRAINT WAREHOUSE64CKC;
ALTER TABLE WAREHOUSE64 ADD CONSTRAINT WAREHOUSE64CKC CHECK
(W_ID BETWEEN 32761 AND 33280);
SET INTEGRITY FOR WAREHOUSE64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE65 OFF;
ALTER TABLE WAREHOUSE65 DROP CONSTRAINT WAREHOUSE65CKC;
ALTER TABLE WAREHOUSE65 ADD CONSTRAINT WAREHOUSE65CKC CHECK
(W_ID BETWEEN 33281 AND 33800);
SET INTEGRITY FOR WAREHOUSE65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE66 OFF;
ALTER TABLE WAREHOUSE66 DROP CONSTRAINT WAREHOUSE66CKC;
ALTER TABLE WAREHOUSE66 ADD CONSTRAINT WAREHOUSE66CKC CHECK
(W_ID BETWEEN 33801 AND 34320);
SET INTEGRITY FOR WAREHOUSE66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE67 OFF;
ALTER TABLE WAREHOUSE67 DROP CONSTRAINT WAREHOUSE67CKC;
ALTER TABLE WAREHOUSE67 ADD CONSTRAINT WAREHOUSE67CKC CHECK
(W_ID BETWEEN 34321 AND 34840);
SET INTEGRITY FOR WAREHOUSE67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE68 OFF;
ALTER TABLE WAREHOUSE68 DROP CONSTRAINT WAREHOUSE68CKC;
ALTER TABLE WAREHOUSE68 ADD CONSTRAINT WAREHOUSE68CKC CHECK
(W_ID BETWEEN 34841 AND 35360);
SET INTEGRITY FOR WAREHOUSE68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE69 OFF;
ALTER TABLE WAREHOUSE69 DROP CONSTRAINT WAREHOUSE69CKC;
ALTER TABLE WAREHOUSE69 ADD CONSTRAINT WAREHOUSE69CKC CHECK
(W_ID BETWEEN 35361 AND 35880);
SET INTEGRITY FOR WAREHOUSE69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE70 OFF;
ALTER TABLE WAREHOUSE70 DROP CONSTRAINT WAREHOUSE70CKC;
ALTER TABLE WAREHOUSE70 ADD CONSTRAINT WAREHOUSE70CKC CHECK
(W_ID BETWEEN 35881 AND 36400);
SET INTEGRITY FOR WAREHOUSE70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE71 OFF;
ALTER TABLE WAREHOUSE71 DROP CONSTRAINT WAREHOUSE71CKC;
ALTER TABLE WAREHOUSE71 ADD CONSTRAINT WAREHOUSE71CKC CHECK
(W_ID BETWEEN 36401 AND 36920);
SET INTEGRITY FOR WAREHOUSE71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE72 OFF;
ALTER TABLE WAREHOUSE72 DROP CONSTRAINT WAREHOUSE72CKC;
ALTER TABLE WAREHOUSE72 ADD CONSTRAINT WAREHOUSE72CKC CHECK
(W_ID BETWEEN 36921 AND 37440);
SET INTEGRITY FOR WAREHOUSE72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE73 OFF;
ALTER TABLE WAREHOUSE73 DROP CONSTRAINT WAREHOUSE73CKC;
ALTER TABLE WAREHOUSE73 ADD CONSTRAINT WAREHOUSE73CKC CHECK
(W_ID BETWEEN 37441 AND 37960);
SET INTEGRITY FOR WAREHOUSE73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE74 OFF;
ALTER TABLE WAREHOUSE74 DROP CONSTRAINT WAREHOUSE74CKC;
ALTER TABLE WAREHOUSE74 ADD CONSTRAINT WAREHOUSE74CKC CHECK
(W_ID BETWEEN 37961 AND 38480);
SET INTEGRITY FOR WAREHOUSE74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE75 OFF;
ALTER TABLE WAREHOUSE75 DROP CONSTRAINT WAREHOUSE75CKC;
ALTER TABLE WAREHOUSE75 ADD CONSTRAINT WAREHOUSE75CKC CHECK
(W_ID BETWEEN 38481 AND 39000);
SET INTEGRITY FOR WAREHOUSE75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE76 OFF;
ALTER TABLE WAREHOUSE76 DROP CONSTRAINT WAREHOUSE76CKC;
ALTER TABLE WAREHOUSE76 ADD CONSTRAINT WAREHOUSE76CKC CHECK
(W_ID BETWEEN 39001 AND 39520);
SET INTEGRITY FOR WAREHOUSE76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE77 OFF;
ALTER TABLE WAREHOUSE77 DROP CONSTRAINT WAREHOUSE77CKC;
ALTER TABLE WAREHOUSE77 ADD CONSTRAINT WAREHOUSE77CKC CHECK
(W_ID BETWEEN 39521 AND 40040);
SET INTEGRITY FOR WAREHOUSE77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE78 OFF;
ALTER TABLE WAREHOUSE78 DROP CONSTRAINT WAREHOUSE78CKC;
ALTER TABLE WAREHOUSE78 ADD CONSTRAINT WAREHOUSE78CKC CHECK
(W_ID BETWEEN 40041 AND 40560);
SET INTEGRITY FOR WAREHOUSE78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE79 OFF;
ALTER TABLE WAREHOUSE79 DROP CONSTRAINT WAREHOUSE79CKC;
ALTER TABLE WAREHOUSE79 ADD CONSTRAINT WAREHOUSE79CKC CHECK
(W_ID BETWEEN 40561 AND 41080);
SET INTEGRITY FOR WAREHOUSE79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE80 OFF;
ALTER TABLE WAREHOUSE80 DROP CONSTRAINT WAREHOUSE80CKC;
ALTER TABLE WAREHOUSE80 ADD CONSTRAINT WAREHOUSE80CKC CHECK
(W_ID >= 41081);
SET INTEGRITY FOR WAREHOUSE80 ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRIDX_CUST_IDXB.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB1;
CREATE INDEX CUST_IDXB1
                ON CUSTOMER1(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
```

```
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB2;
CREATE INDEX CUST_IDXB2
        ON CUSTOMER2(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB3;
CREATE INDEX CUST_IDXB3
        ON CUSTOMER3(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB4;
CREATE INDEX CUST_IDXB4
        ON CUSTOMER4(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB5;
CREATE INDEX CUST_IDXB5
        ON CUSTOMER5(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB6;
CREATE INDEX CUST_IDXB6
        ON CUSTOMER6(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB7;
CREATE INDEX CUST_IDXB7
        ON CUSTOMER7(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB8;
CREATE INDEX CUST_IDXB8
        ON CUSTOMER8(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB9;
CREATE INDEX CUST_IDXB9
        ON CUSTOMER9(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB10;
CREATE INDEX CUST_IDXB10
        ON CUSTOMER10(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB11;
CREATE INDEX CUST_IDXB11
        ON CUSTOMER11(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB12;
CREATE INDEX CUST_IDXB12
        ON CUSTOMER12(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB13;
CREATE INDEX CUST_IDXB13
        ON CUSTOMER13(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB14;
CREATE INDEX CUST_IDXB14
        ON CUSTOMER14(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB15;
CREATE INDEX CUST_IDXB15
        ON CUSTOMER15(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB16;
CREATE INDEX CUST_IDXB16
        ON CUSTOMER16(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB17;
CREATE INDEX CUST_IDXB17
        ON CUSTOMER17(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB18;
CREATE INDEX CUST_IDXB18
        ON CUSTOMER18(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB19;
CREATE INDEX CUST_IDXB19
        ON CUSTOMER19(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB20;
CREATE INDEX CUST_IDXB20
        ON CUSTOMER20(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB21;
CREATE INDEX CUST_IDXB21
        ON CUSTOMER21(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB22;
CREATE INDEX CUST_IDXB22
        ON CUSTOMER22(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB23;
CREATE INDEX CUST_IDXB23
        ON CUSTOMER23(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB24;
CREATE INDEX CUST_IDXB24
        ON CUSTOMER24(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB25;
CREATE INDEX CUST_IDXB25
        ON CUSTOMER25(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB26;
CREATE INDEX CUST_IDXB26
        ON CUSTOMER26(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB27;
CREATE INDEX CUST_IDXB27
        ON CUSTOMER27(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB28;
CREATE INDEX CUST_IDXB28
        ON CUSTOMER28(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB29;
CREATE INDEX CUST_IDXB29
        ON CUSTOMER29(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB30;
CREATE INDEX CUST_IDXB30
        ON CUSTOMER30(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB31;
CREATE INDEX CUST_IDXB31
        ON CUSTOMER31(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB32;
CREATE INDEX CUST_IDXB32
        ON CUSTOMER32(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB33;
CREATE INDEX CUST_IDXB33
        ON CUSTOMER33(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB34;
CREATE INDEX CUST_IDXB34
        ON CUSTOMER34(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB35;
CREATE INDEX CUST_IDXB35
        ON CUSTOMER35(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB36;
CREATE INDEX CUST_IDXB36
        ON CUSTOMER36(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
```

```
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB37;
CREATE INDEX CUST_IDXB37
        ON CUSTOMER37(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB38;
CREATE INDEX CUST_IDXB38
        ON CUSTOMER38(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB39;
CREATE INDEX CUST_IDXB39
        ON CUSTOMER39(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB40;
CREATE INDEX CUST_IDXB40
        ON CUSTOMER40(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB41;
CREATE INDEX CUST_IDXB41
        ON CUSTOMER41(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB42;
CREATE INDEX CUST_IDXB42
        ON CUSTOMER42(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB43;
CREATE INDEX CUST_IDXB43
        ON CUSTOMER43(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB44;
CREATE INDEX CUST_IDXB44
        ON CUSTOMER44(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB45;
CREATE INDEX CUST_IDXB45
        ON CUSTOMER45(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB46;
CREATE INDEX CUST_IDXB46
        ON CUSTOMER46(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB47;
CREATE INDEX CUST_IDXB47
        ON CUSTOMER47(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB48;
CREATE INDEX CUST_IDXB48
        ON CUSTOMER48(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB49;
CREATE INDEX CUST_IDXB49
        ON CUSTOMER49(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB50;
CREATE INDEX CUST_IDXB50
        ON CUSTOMER50(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB51;
CREATE INDEX CUST_IDXB51
        ON CUSTOMER51(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB52;
CREATE INDEX CUST_IDXB52
        ON CUSTOMER52(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB53;
CREATE INDEX CUST_IDXB53
        ON CUSTOMER53(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB54;
CREATE INDEX CUST_IDXB54
        ON CUSTOMER54(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB55;
CREATE INDEX CUST_IDXB55
        ON CUSTOMER55(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB56;
CREATE INDEX CUST_IDXB56
        ON CUSTOMER56(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB57;
CREATE INDEX CUST_IDXB57
        ON CUSTOMER57(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB58;
CREATE INDEX CUST_IDXB58
        ON CUSTOMER58(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB59;
CREATE INDEX CUST_IDXB59
        ON CUSTOMER59(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB60;
CREATE INDEX CUST_IDXB60
        ON CUSTOMER60(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB61;
CREATE INDEX CUST_IDXB61
        ON CUSTOMER61(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB62;
CREATE INDEX CUST_IDXB62
        ON CUSTOMER62(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB63;
CREATE INDEX CUST_IDXB63
        ON CUSTOMER63(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB64;
CREATE INDEX CUST_IDXB64
        ON CUSTOMER64(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB65;
CREATE INDEX CUST_IDXB65
        ON CUSTOMER65(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB66;
CREATE INDEX CUST_IDXB66
        ON CUSTOMER66(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB67;
CREATE INDEX CUST_IDXB67
        ON CUSTOMER67(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB68;
CREATE INDEX CUST_IDXB68
        ON CUSTOMER68(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB69;
CREATE INDEX CUST_IDXB69
        ON CUSTOMER69(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB70;
CREATE INDEX CUST_IDXB70
        ON CUSTOMER70(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB71;
CREATE INDEX CUST_IDXB71
        ON CUSTOMER71(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
```

```
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB72;
CREATE INDEX CUST_IDXB72
        ON CUSTOMER72(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB73;
CREATE INDEX CUST_IDXB73
        ON CUSTOMER73(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB74;
CREATE INDEX CUST_IDXB74
        ON CUSTOMER74(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB75;
CREATE INDEX CUST_IDXB75
        ON CUSTOMER75(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB76;
CREATE INDEX CUST_IDXB76
        ON CUSTOMER76(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB77;
CREATE INDEX CUST_IDXB77
        ON CUSTOMER77(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB78;
CREATE INDEX CUST_IDXB78
        ON CUSTOMER78(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB79;
CREATE INDEX CUST_IDXB79
        ON CUSTOMER79(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB80;
CREATE INDEX CUST_IDXB80
        ON CUSTOMER80(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID)
PCTFREE 0;
connect reset;
```

DDL/CRIDX_ORDR_IDXB.ddl

```
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB1;
CREATE INDEX ORDR_IDXB1
        ON ORDERS1(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB2;
CREATE INDEX ORDR_IDXB2
        ON ORDERS2(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB3;
CREATE INDEX ORDR_IDXB3
        ON ORDERS3(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB4;
CREATE INDEX ORDR_IDXB4
        ON ORDERS4(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB5;
CREATE INDEX ORDR_IDXB5
        ON ORDERS5(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB6;
CREATE INDEX ORDR_IDXB6
        ON ORDERS6(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB7;
CREATE INDEX ORDR_IDXB7
        ON ORDERS7(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB8;
CREATE INDEX ORDR_IDXB8
        ON ORDERS8(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB9;
CREATE INDEX ORDR_IDXB9
        ON ORDERS9(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB10;
CREATE INDEX ORDR_IDXB10
        ON ORDERS10(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB11;
CREATE INDEX ORDR_IDXB11
        ON ORDERS11(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB12;
CREATE INDEX ORDR_IDXB12
        ON ORDERS12(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB13;
CREATE INDEX ORDR_IDXB13
        ON ORDERS13(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB14;
CREATE INDEX ORDR_IDXB14
        ON ORDERS14(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB15;
CREATE INDEX ORDR_IDXB15
        ON ORDERS15(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB16;
CREATE INDEX ORDR_IDXB16
        ON ORDERS16(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB17;
CREATE INDEX ORDR_IDXB17
        ON ORDERS17(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB18;
CREATE INDEX ORDR_IDXB18
        ON ORDERS18(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB19;
CREATE INDEX ORDR_IDXB19
        ON ORDERS19(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB20;
CREATE INDEX ORDR_IDXB20
        ON ORDERS20(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB21;
CREATE INDEX ORDR_IDXB21
        ON ORDERS21(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB22;
CREATE INDEX ORDR_IDXB22
        ON ORDERS22(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB23;
CREATE INDEX ORDR_IDXB23
        ON ORDERS23(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB24;
CREATE INDEX ORDR_IDXB24
        ON ORDERS24(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB25;
CREATE INDEX ORDR_IDXB25
        ON ORDERS25(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
```

```
DROP INDEX ORDR_IDXB26;
CREATE INDEX ORDR_IDXB26
        ON ORDERS26(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB27;
CREATE INDEX ORDR_IDXB27
        ON ORDERS27(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB28;
CREATE INDEX ORDR_IDXB28
        ON ORDERS28(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB29;
CREATE INDEX ORDR_IDXB29
        ON ORDERS29(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB30;
CREATE INDEX ORDR_IDXB30
        ON ORDERS30(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB31;
CREATE INDEX ORDR_IDXB31
        ON ORDERS31(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB32;
CREATE INDEX ORDR_IDXB32
        ON ORDERS32(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB33;
CREATE INDEX ORDR_IDXB33
        ON ORDERS33(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB34;
CREATE INDEX ORDR_IDXB34
        ON ORDERS34(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB35;
CREATE INDEX ORDR_IDXB35
        ON ORDERS35(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB36;
CREATE INDEX ORDR_IDXB36
        ON ORDERS36(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB37;
CREATE INDEX ORDR_IDXB37
        ON ORDERS37(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB38;
CREATE INDEX ORDR_IDXB38
        ON ORDERS38(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB39;
CREATE INDEX ORDR_IDXB39
        ON ORDERS39(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB40;
CREATE INDEX ORDR_IDXB40
        ON ORDERS40(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB41;
CREATE INDEX ORDR_IDXB41
        ON ORDERS41(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB42;
CREATE INDEX ORDR_IDXB42
        ON ORDERS42(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB43;
CREATE INDEX ORDR_IDXB43
        ON ORDERS43(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB44;
CREATE INDEX ORDR_IDXB44
        ON ORDERS44(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB45;
CREATE INDEX ORDR_IDXB45
        ON ORDERS45(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB46;
CREATE INDEX ORDR_IDXB46
        ON ORDERS46(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB47;
CREATE INDEX ORDR_IDXB47
        ON ORDERS47(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB48;
CREATE INDEX ORDR_IDXB48
        ON ORDERS48(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB49;
CREATE INDEX ORDR_IDXB49
        ON ORDERS49(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB50;
CREATE INDEX ORDR_IDXB50
        ON ORDERS50(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB51;
CREATE INDEX ORDR_IDXB51
        ON ORDERS51(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB52;
CREATE INDEX ORDR_IDXB52
        ON ORDERS52(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB53;
CREATE INDEX ORDR_IDXB53
        ON ORDERS53(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB54;
CREATE INDEX ORDR_IDXB54
        ON ORDERS54(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB55;
CREATE INDEX ORDR_IDXB55
        ON ORDERS55(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB56;
CREATE INDEX ORDR_IDXB56
        ON ORDERS56(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB57;
CREATE INDEX ORDR_IDXB57
        ON ORDERS57(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB58;
CREATE INDEX ORDR_IDXB58
        ON ORDERS58(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB59;
CREATE INDEX ORDR_IDXB59
        ON ORDERS59(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB60;
CREATE INDEX ORDR_IDXB60
        ON ORDERS60(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
```

```
DROP INDEX ORDR_IDXB61;
CREATE INDEX ORDR_IDXB61
        ON ORDERS61(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB62;
CREATE INDEX ORDR_IDXB62
        ON ORDERS62(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB63;
CREATE INDEX ORDR_IDXB63
        ON ORDERS63(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB64;
CREATE INDEX ORDR_IDXB64
        ON ORDERS64(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB65;
CREATE INDEX ORDR_IDXB65
        ON ORDERS65(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB66;
CREATE INDEX ORDR_IDXB66
        ON ORDERS66(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB67;
CREATE INDEX ORDR_IDXB67
        ON ORDERS67(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB68;
CREATE INDEX ORDR_IDXB68
        ON ORDERS68(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB69;
CREATE INDEX ORDR_IDXB69
        ON ORDERS69(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB70;
CREATE INDEX ORDR_IDXB70
        ON ORDERS70(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB71;
CREATE INDEX ORDR_IDXB71
        ON ORDERS71(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB72;
CREATE INDEX ORDR_IDXB72
        ON ORDERS72(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB73;
CREATE INDEX ORDR_IDXB73
        ON ORDERS73(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB74;
CREATE INDEX ORDR_IDXB74
        ON ORDERS74(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB75;
CREATE INDEX ORDR_IDXB75
        ON ORDERS75(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB76;
CREATE INDEX ORDR_IDXB76
        ON ORDERS76(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB77;
CREATE INDEX ORDR_IDXB77
        ON ORDERS77(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB78;
CREATE INDEX ORDR_IDXB78
        ON ORDERS78(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB79;
CREATE INDEX ORDR_IDXB79
        ON ORDERS79(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB80;
CREATE INDEX ORDR_IDXB80
        ON ORDERS80(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
```

DDL/CRTB_CUSTOMER.ddl

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER1;
CREATE TABLE CUSTOMER1
        (
         C_ID          INTEGER       NOT NULL,
         C_STATE       CHAR(2)       NOT NULL,
         C_ZIP         CHAR(9)       NOT NULL,
         C_PHONE       CHAR(16)      NOT NULL,
         C_SINCE       TIMESTAMP     NOT NULL,
         C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
         C_MIDDLE      CHAR(2)       NOT NULL,
         C_CREDIT      CHAR(2)       NOT NULL,
         C_DISCOUNT    REAL          NOT NULL,
         C_DATA        VARCHAR(500)  NOT NULL,
         C_LAST        VARCHAR(16)   NOT NULL,
         C_FIRST       VARCHAR(16)   NOT NULL,
         C_STREET_1    VARCHAR(20)   NOT NULL,
         C_STREET_2    VARCHAR(20)   NOT NULL,
         C_CITY        VARCHAR(20)   NOT NULL,
         C_D_ID        SMALLINT      NOT NULL,
         C_W_ID        INTEGER       NOT NULL,
         C_DELIVERY_CNT INTEGER      NOT NULL,
         C_BALANCE     DECIMAL(12,2) NOT NULL,
         C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
         C_PAYMENT_CNT INTEGER       NOT NULL
        )
        IN ts_customer_001
        INDEX IN is_customer_001
        ORGANIZE BY KEY SEQUENCE (
         C_ID STARTING FROM 1 ENDING AT 3000,
         C_W_ID STARTING FROM 1 ENDING AT 520,
         C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER2;
CREATE TABLE CUSTOMER2
        (
         C_ID          INTEGER       NOT NULL,
         C_STATE       CHAR(2)       NOT NULL,
         C_ZIP         CHAR(9)       NOT NULL,
         C_PHONE       CHAR(16)      NOT NULL,
         C_SINCE       TIMESTAMP     NOT NULL,
         C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
         C_MIDDLE      CHAR(2)       NOT NULL,
         C_CREDIT      CHAR(2)       NOT NULL,
         C_DISCOUNT    REAL          NOT NULL,
         C_DATA        VARCHAR(500)  NOT NULL,
         C_LAST        VARCHAR(16)   NOT NULL,
         C_FIRST       VARCHAR(16)   NOT NULL,
         C_STREET_1    VARCHAR(20)   NOT NULL,
         C_STREET_2    VARCHAR(20)   NOT NULL,
         C_CITY        VARCHAR(20)   NOT NULL,
         C_D_ID        SMALLINT      NOT NULL,
         C_W_ID        INTEGER       NOT NULL,
         C_DELIVERY_CNT INTEGER      NOT NULL,
         C_BALANCE     DECIMAL(12,2) NOT NULL,
         C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
         C_PAYMENT_CNT INTEGER       NOT NULL
        )
        IN ts_customer_002
        INDEX IN is_customer_002
        ORGANIZE BY KEY SEQUENCE (
         C_ID STARTING FROM 1 ENDING AT 3000,
         C_W_ID STARTING FROM 521 ENDING AT 1040,
         C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER3;
CREATE TABLE CUSTOMER3
        (
         C_ID          INTEGER       NOT NULL,
         C_STATE       CHAR(2)       NOT NULL,
         C_ZIP         CHAR(9)       NOT NULL,
         C_PHONE       CHAR(16)      NOT NULL,
         C_SINCE       TIMESTAMP     NOT NULL,
         C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
         C_MIDDLE      CHAR(2)       NOT NULL,
         C_CREDIT      CHAR(2)       NOT NULL,
         C_DISCOUNT    REAL          NOT NULL,
         C_DATA        VARCHAR(500)  NOT NULL,
         C_LAST        VARCHAR(16)   NOT NULL,
         C_FIRST       VARCHAR(16)   NOT NULL,
         C_STREET_1    VARCHAR(20)   NOT NULL,
```

```
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_003
            INDEX IN is_customer_003
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 1041 ENDING AT 1560,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER4;
CREATE TABLE CUSTOMER4
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_004
            INDEX IN is_customer_004
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 1561 ENDING AT 2080,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER5;
CREATE TABLE CUSTOMER5
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_005
            INDEX IN is_customer_005
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 2081 ENDING AT 2600,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER6;
CREATE TABLE CUSTOMER6
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_006
            INDEX IN is_customer_006
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 2601 ENDING AT 3120,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER7;
CREATE TABLE CUSTOMER7
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_007
            INDEX IN is_customer_007
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 3121 ENDING AT 3640,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER8;
CREATE TABLE CUSTOMER8
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_008
            INDEX IN is_customer_008
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 3641 ENDING AT 4160,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER9;
CREATE TABLE CUSTOMER9
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_009
INDEX IN is_customer_009
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 4161 ENDING AT 4680,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER10;
CREATE TABLE CUSTOMER10
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_010
INDEX IN is_customer_010
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 4681 ENDING AT 5200,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER11;
CREATE TABLE CUSTOMER11
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_011
INDEX IN is_customer_011
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 5201 ENDING AT 5720,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER12;
CREATE TABLE CUSTOMER12
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_012
INDEX IN is_customer_012
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 5721 ENDING AT 6240,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER13;
CREATE TABLE CUSTOMER13
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_013
INDEX IN is_customer_013
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 6241 ENDING AT 6760,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER14;
CREATE TABLE CUSTOMER14
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_014
INDEX IN is_customer_014
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 6761 ENDING AT 7280,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER15;
CREATE TABLE CUSTOMER15
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_015
INDEX IN is_customer_015
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 7281 ENDING AT 7800,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER16;
CREATE TABLE CUSTOMER16
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_016
INDEX IN is_customer_016
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 7801 ENDING AT 8320,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER17;
CREATE TABLE CUSTOMER17
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_017
INDEX IN is_customer_017
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 8321 ENDING AT 8840,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER18;
CREATE TABLE CUSTOMER18
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_018
INDEX IN is_customer_018
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 8841 ENDING AT 9360,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER19;
CREATE TABLE CUSTOMER19
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_019
INDEX IN is_customer_019
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 9361 ENDING AT 9880,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER20;
CREATE TABLE CUSTOMER20
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_020
INDEX IN is_customer_020
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 9881 ENDING AT 10400,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER21;
CREATE TABLE CUSTOMER21
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_021
            INDEX IN is_customer_021
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 10401 ENDING AT 10920,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER22;
CREATE TABLE CUSTOMER22
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_022
            INDEX IN is_customer_022
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 10921 ENDING AT 11440,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER23;
CREATE TABLE CUSTOMER23
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_023
            INDEX IN is_customer_023
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 11441 ENDING AT 11960,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER24;
CREATE TABLE CUSTOMER24
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_024
            INDEX IN is_customer_024
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 11961 ENDING AT 12480,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER25;
CREATE TABLE CUSTOMER25
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_025
            INDEX IN is_customer_025
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 12481 ENDING AT 13000,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER26;
CREATE TABLE CUSTOMER26
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_026
            INDEX IN is_customer_026
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 13001 ENDING AT 13520,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER27;
CREATE TABLE CUSTOMER27
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_027
INDEX IN is_customer_027
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 13521 ENDING AT 14040,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER28;
CREATE TABLE CUSTOMER28
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_028
INDEX IN is_customer_028
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 14041 ENDING AT 14560,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER29;
CREATE TABLE CUSTOMER29
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_029
INDEX IN is_customer_029
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 14561 ENDING AT 15080,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER30;
CREATE TABLE CUSTOMER30
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_030
INDEX IN is_customer_030
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 15081 ENDING AT 15600,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER31;
CREATE TABLE CUSTOMER31
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_031
INDEX IN is_customer_031
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 15601 ENDING AT 16120,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER32;
CREATE TABLE CUSTOMER32
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_032
INDEX IN is_customer_032
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 16121 ENDING AT 16640,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER33;
CREATE TABLE CUSTOMER33
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
```

```
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_033
            INDEX IN is_customer_033
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 16641 ENDING AT 17160,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER34;
CREATE TABLE CUSTOMER34
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_034
            INDEX IN is_customer_034
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 17161 ENDING AT 17680,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER35;
CREATE TABLE CUSTOMER35
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_035
            INDEX IN is_customer_035
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 17681 ENDING AT 18200,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER36;
CREATE TABLE CUSTOMER36
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_036
            INDEX IN is_customer_036
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 18201 ENDING AT 18720,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER37;
CREATE TABLE CUSTOMER37
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_037
            INDEX IN is_customer_037
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 18721 ENDING AT 19240,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER38;
CREATE TABLE CUSTOMER38
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_038
            INDEX IN is_customer_038
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 19241 ENDING AT 19760,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER39;
CREATE TABLE CUSTOMER39
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_039
INDEX IN is_customer_039
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 19761 ENDING AT 20280,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER40;
CREATE TABLE CUSTOMER40
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_040
INDEX IN is_customer_040
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 20281 ENDING AT 20800,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER41;
CREATE TABLE CUSTOMER41
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_041
INDEX IN is_customer_041
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 20801 ENDING AT 21320,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER42;
CREATE TABLE CUSTOMER42
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_042
INDEX IN is_customer_042
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 21321 ENDING AT 21840,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER43;
CREATE TABLE CUSTOMER43
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_043
INDEX IN is_customer_043
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 21841 ENDING AT 22360,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER44;
CREATE TABLE CUSTOMER44
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_044
INDEX IN is_customer_044
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 22361 ENDING AT 22880,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER45;
CREATE TABLE CUSTOMER45
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_045
INDEX IN is_customer_045
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 22881 ENDING AT 23400,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER46;
CREATE TABLE CUSTOMER46
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_046
INDEX IN is_customer_046
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 23401 ENDING AT 23920,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER47;
CREATE TABLE CUSTOMER47
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_047
INDEX IN is_customer_047
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 23921 ENDING AT 24440,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER48;
CREATE TABLE CUSTOMER48
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_048
INDEX IN is_customer_048
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 24441 ENDING AT 24960,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER49;
CREATE TABLE CUSTOMER49
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_049
INDEX IN is_customer_049
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 24961 ENDING AT 25480,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER50;
CREATE TABLE CUSTOMER50
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_050
INDEX IN is_customer_050
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 25481 ENDING AT 26000,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER51;
CREATE TABLE CUSTOMER51
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
 C_MIDDLE        CHAR(2)       NOT NULL,
 C_CREDIT        CHAR(2)       NOT NULL,
 C_DISCOUNT      REAL          NOT NULL,
 C_DATA          VARCHAR(500)  NOT NULL,
 C_LAST          VARCHAR(16)   NOT NULL,
 C_FIRST         VARCHAR(16)   NOT NULL,
 C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_051
INDEX IN is_customer_051
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 26001 ENDING AT 26520,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER52;
CREATE TABLE CUSTOMER52
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_052
INDEX IN is_customer_052
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 26521 ENDING AT 27040,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER53;
CREATE TABLE CUSTOMER53
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_053
INDEX IN is_customer_053
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 27041 ENDING AT 27560,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER54;
CREATE TABLE CUSTOMER54
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_054
INDEX IN is_customer_054
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 27561 ENDING AT 28080,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER55;
CREATE TABLE CUSTOMER55
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_055
INDEX IN is_customer_055
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 28081 ENDING AT 28600,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER56;
CREATE TABLE CUSTOMER56
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_056
INDEX IN is_customer_056
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 28601 ENDING AT 29120,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER57;
CREATE TABLE CUSTOMER57
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
```

```
 C_STREET_2      VARCHAR(20)   NOT NULL,
 C_CITY          VARCHAR(20)   NOT NULL,
 C_D_ID          SMALLINT      NOT NULL,
 C_W_ID          INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_057
INDEX IN is_customer_057
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 29121 ENDING AT 29640,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER58;
CREATE TABLE CUSTOMER58
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_058
INDEX IN is_customer_058
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 29641 ENDING AT 30160,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER59;
CREATE TABLE CUSTOMER59
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_059
INDEX IN is_customer_059
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 30161 ENDING AT 30680,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER60;
CREATE TABLE CUSTOMER60
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_060
INDEX IN is_customer_060
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 30681 ENDING AT 31200,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER61;
CREATE TABLE CUSTOMER61
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_061
INDEX IN is_customer_061
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 31201 ENDING AT 31720,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER62;
CREATE TABLE CUSTOMER62
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
 C_STREET_2     VARCHAR(20)   NOT NULL,
 C_CITY         VARCHAR(20)   NOT NULL,
 C_D_ID         SMALLINT      NOT NULL,
 C_W_ID         INTEGER       NOT NULL,
 C_DELIVERY_CNT  INTEGER       NOT NULL,
 C_BALANCE       DECIMAL(12,2) NOT NULL,
 C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
 C_PAYMENT_CNT   INTEGER       NOT NULL
)
IN ts_customer_062
INDEX IN is_customer_062
ORGANIZE BY KEY SEQUENCE (
 C_ID STARTING FROM 1 ENDING AT 3000,
 C_W_ID STARTING FROM 31721 ENDING AT 32240,
 C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER63;
CREATE TABLE CUSTOMER63
(
 C_ID           INTEGER       NOT NULL,
 C_STATE        CHAR(2)       NOT NULL,
 C_ZIP          CHAR(9)       NOT NULL,
 C_PHONE        CHAR(16)      NOT NULL,
 C_SINCE        TIMESTAMP     NOT NULL,
 C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
 C_MIDDLE       CHAR(2)       NOT NULL,
 C_CREDIT       CHAR(2)       NOT NULL,
 C_DISCOUNT     REAL          NOT NULL,
 C_DATA         VARCHAR(500)  NOT NULL,
 C_LAST         VARCHAR(16)   NOT NULL,
 C_FIRST        VARCHAR(16)   NOT NULL,
 C_STREET_1     VARCHAR(20)   NOT NULL,
```

```
    C_STREET_2      VARCHAR(20)   NOT NULL,
    C_CITY          VARCHAR(20)   NOT NULL,
    C_D_ID          SMALLINT      NOT NULL,
    C_W_ID          INTEGER       NOT NULL,
    C_DELIVERY_CNT  INTEGER       NOT NULL,
    C_BALANCE       DECIMAL(12,2) NOT NULL,
    C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
    C_PAYMENT_CNT   INTEGER       NOT NULL
    )
    IN ts_customer_063
    INDEX IN is_customer_063
    ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 32241 ENDING AT 32760,
    C_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER64;
CREATE TABLE CUSTOMER64
    (
    C_ID           INTEGER       NOT NULL,
    C_STATE        CHAR(2)       NOT NULL,
    C_ZIP          CHAR(9)       NOT NULL,
    C_PHONE        CHAR(16)      NOT NULL,
    C_SINCE        TIMESTAMP     NOT NULL,
    C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
    C_MIDDLE       CHAR(2)       NOT NULL,
    C_CREDIT       CHAR(2)       NOT NULL,
    C_DISCOUNT     REAL          NOT NULL,
    C_DATA         VARCHAR(500)  NOT NULL,
    C_LAST         VARCHAR(16)   NOT NULL,
    C_FIRST        VARCHAR(16)   NOT NULL,
    C_STREET_1     VARCHAR(20)   NOT NULL,
    C_STREET_2     VARCHAR(20)   NOT NULL,
    C_CITY         VARCHAR(20)   NOT NULL,
    C_D_ID         SMALLINT      NOT NULL,
    C_W_ID         INTEGER       NOT NULL,
    C_DELIVERY_CNT  INTEGER      NOT NULL,
    C_BALANCE       DECIMAL(12,2) NOT NULL,
    C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
    C_PAYMENT_CNT   INTEGER      NOT NULL
    )
    IN ts_customer_064
    INDEX IN is_customer_064
    ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 32761 ENDING AT 33280,
    C_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER65;
CREATE TABLE CUSTOMER65
    (
    C_ID           INTEGER       NOT NULL,
    C_STATE        CHAR(2)       NOT NULL,
    C_ZIP          CHAR(9)       NOT NULL,
    C_PHONE        CHAR(16)      NOT NULL,
    C_SINCE        TIMESTAMP     NOT NULL,
    C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
    C_MIDDLE       CHAR(2)       NOT NULL,
    C_CREDIT       CHAR(2)       NOT NULL,
    C_DISCOUNT     REAL          NOT NULL,
    C_DATA         VARCHAR(500)  NOT NULL,
    C_LAST         VARCHAR(16)   NOT NULL,
    C_FIRST        VARCHAR(16)   NOT NULL,
    C_STREET_1     VARCHAR(20)   NOT NULL,
    C_STREET_2     VARCHAR(20)   NOT NULL,
    C_CITY         VARCHAR(20)   NOT NULL,
    C_D_ID         SMALLINT      NOT NULL,
    C_W_ID         INTEGER       NOT NULL,
    C_DELIVERY_CNT  INTEGER      NOT NULL,
    C_BALANCE       DECIMAL(12,2) NOT NULL,
    C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
    C_PAYMENT_CNT   INTEGER      NOT NULL
    )
    IN ts_customer_065
    INDEX IN is_customer_065
    ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 33281 ENDING AT 33800,
    C_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER66;
CREATE TABLE CUSTOMER66
    (
    C_ID           INTEGER       NOT NULL,
    C_STATE        CHAR(2)       NOT NULL,
    C_ZIP          CHAR(9)       NOT NULL,
    C_PHONE        CHAR(16)      NOT NULL,
    C_SINCE        TIMESTAMP     NOT NULL,
    C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
    C_MIDDLE       CHAR(2)       NOT NULL,
    C_CREDIT       CHAR(2)       NOT NULL,
    C_DISCOUNT     REAL          NOT NULL,
    C_DATA         VARCHAR(500)  NOT NULL,
    C_LAST         VARCHAR(16)   NOT NULL,
    C_FIRST        VARCHAR(16)   NOT NULL,
    C_STREET_1     VARCHAR(20)   NOT NULL,
    C_STREET_2     VARCHAR(20)   NOT NULL,
    C_CITY         VARCHAR(20)   NOT NULL,
    C_D_ID         SMALLINT      NOT NULL,
    C_W_ID         INTEGER       NOT NULL,
    C_DELIVERY_CNT  INTEGER      NOT NULL,
    C_BALANCE       DECIMAL(12,2) NOT NULL,
    C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
    C_PAYMENT_CNT   INTEGER      NOT NULL
    )
    IN ts_customer_066
    INDEX IN is_customer_066
    ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 33801 ENDING AT 34320,
    C_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER67;
CREATE TABLE CUSTOMER67
    (
    C_ID           INTEGER       NOT NULL,
    C_STATE        CHAR(2)       NOT NULL,
    C_ZIP          CHAR(9)       NOT NULL,
    C_PHONE        CHAR(16)      NOT NULL,
    C_SINCE        TIMESTAMP     NOT NULL,
    C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
    C_MIDDLE       CHAR(2)       NOT NULL,
    C_CREDIT       CHAR(2)       NOT NULL,
    C_DISCOUNT     REAL          NOT NULL,
    C_DATA         VARCHAR(500)  NOT NULL,
    C_LAST         VARCHAR(16)   NOT NULL,
    C_FIRST        VARCHAR(16)   NOT NULL,
    C_STREET_1     VARCHAR(20)   NOT NULL,
    C_STREET_2     VARCHAR(20)   NOT NULL,
    C_CITY         VARCHAR(20)   NOT NULL,
    C_D_ID         SMALLINT      NOT NULL,
    C_W_ID         INTEGER       NOT NULL,
    C_DELIVERY_CNT  INTEGER      NOT NULL,
    C_BALANCE       DECIMAL(12,2) NOT NULL,
    C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
    C_PAYMENT_CNT   INTEGER      NOT NULL
    )
    IN ts_customer_067
    INDEX IN is_customer_067
    ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 34321 ENDING AT 34840,
    C_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER68;
CREATE TABLE CUSTOMER68
    (
    C_ID           INTEGER       NOT NULL,
    C_STATE        CHAR(2)       NOT NULL,
    C_ZIP          CHAR(9)       NOT NULL,
    C_PHONE        CHAR(16)      NOT NULL,
    C_SINCE        TIMESTAMP     NOT NULL,
    C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
    C_MIDDLE       CHAR(2)       NOT NULL,
    C_CREDIT       CHAR(2)       NOT NULL,
    C_DISCOUNT     REAL          NOT NULL,
    C_DATA         VARCHAR(500)  NOT NULL,
    C_LAST         VARCHAR(16)   NOT NULL,
    C_FIRST        VARCHAR(16)   NOT NULL,
    C_STREET_1     VARCHAR(20)   NOT NULL,
    C_STREET_2     VARCHAR(20)   NOT NULL,
    C_CITY         VARCHAR(20)   NOT NULL,
    C_D_ID         SMALLINT      NOT NULL,
    C_W_ID         INTEGER       NOT NULL,
    C_DELIVERY_CNT  INTEGER      NOT NULL,
    C_BALANCE       DECIMAL(12,2) NOT NULL,
    C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
    C_PAYMENT_CNT   INTEGER      NOT NULL
    )
    IN ts_customer_068
    INDEX IN is_customer_068
    ORGANIZE BY KEY SEQUENCE (
    C_ID STARTING FROM 1 ENDING AT 3000,
    C_W_ID STARTING FROM 34841 ENDING AT 35360,
    C_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER69;
CREATE TABLE CUSTOMER69
    (
    C_ID           INTEGER       NOT NULL,
    C_STATE        CHAR(2)       NOT NULL,
    C_ZIP          CHAR(9)       NOT NULL,
    C_PHONE        CHAR(16)      NOT NULL,
    C_SINCE        TIMESTAMP     NOT NULL,
    C_CREDIT_LIM   DECIMAL(12,2) NOT NULL,
    C_MIDDLE       CHAR(2)       NOT NULL,
    C_CREDIT       CHAR(2)       NOT NULL,
    C_DISCOUNT     REAL          NOT NULL,
    C_DATA         VARCHAR(500)  NOT NULL,
    C_LAST         VARCHAR(16)   NOT NULL,
    C_FIRST        VARCHAR(16)   NOT NULL,
    C_STREET_1     VARCHAR(20)   NOT NULL,
```

```
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_069
            INDEX IN is_customer_069
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 35361 ENDING AT 35880,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER70;
CREATE TABLE CUSTOMER70
            (
            C_ID           INTEGER       NOT NULL,
            C_STATE        CHAR(2)       NOT NULL,
            C_ZIP          CHAR(9)       NOT NULL,
            C_PHONE        CHAR(16)      NOT NULL,
            C_SINCE        TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_070
            INDEX IN is_customer_070
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 35881 ENDING AT 36400,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER71;
CREATE TABLE CUSTOMER71
            (
            C_ID           INTEGER       NOT NULL,
            C_STATE        CHAR(2)       NOT NULL,
            C_ZIP          CHAR(9)       NOT NULL,
            C_PHONE        CHAR(16)      NOT NULL,
            C_SINCE        TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_071
            INDEX IN is_customer_071
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 36401 ENDING AT 36920,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER72;
CREATE TABLE CUSTOMER72
            (
            C_ID           INTEGER       NOT NULL,
            C_STATE        CHAR(2)       NOT NULL,
            C_ZIP          CHAR(9)       NOT NULL,
            C_PHONE        CHAR(16)      NOT NULL,
            C_SINCE        TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_072
            INDEX IN is_customer_072
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 36921 ENDING AT 37440,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER73;
CREATE TABLE CUSTOMER73
            (
            C_ID           INTEGER       NOT NULL,
            C_STATE        CHAR(2)       NOT NULL,
            C_ZIP          CHAR(9)       NOT NULL,
            C_PHONE        CHAR(16)      NOT NULL,
            C_SINCE        TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_073
            INDEX IN is_customer_073
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 37441 ENDING AT 37960,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER74;
CREATE TABLE CUSTOMER74
            (
            C_ID           INTEGER       NOT NULL,
            C_STATE        CHAR(2)       NOT NULL,
            C_ZIP          CHAR(9)       NOT NULL,
            C_PHONE        CHAR(16)      NOT NULL,
            C_SINCE        TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_074
            INDEX IN is_customer_074
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 37961 ENDING AT 38480,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER75;
CREATE TABLE CUSTOMER75
            (
            C_ID           INTEGER       NOT NULL,
            C_STATE        CHAR(2)       NOT NULL,
            C_ZIP          CHAR(9)       NOT NULL,
            C_PHONE        CHAR(16)      NOT NULL,
            C_SINCE        TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
```

```
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_075
            INDEX IN is_customer_075
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 38481 ENDING AT 39000,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER76;
CREATE TABLE CUSTOMER76
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_076
            INDEX IN is_customer_076
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 39001 ENDING AT 39520,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER77;
CREATE TABLE CUSTOMER77
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_077
            INDEX IN is_customer_077
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 39521 ENDING AT 40040,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER78;
CREATE TABLE CUSTOMER78
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_078
            INDEX IN is_customer_078
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 40041 ENDING AT 40560,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER79;
CREATE TABLE CUSTOMER79
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_079
            INDEX IN is_customer_079
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 40561 ENDING AT 41080,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER80;
CREATE TABLE CUSTOMER80
            (
            C_ID            INTEGER       NOT NULL,
            C_STATE         CHAR(2)       NOT NULL,
            C_ZIP           CHAR(9)       NOT NULL,
            C_PHONE         CHAR(16)      NOT NULL,
            C_SINCE         TIMESTAMP     NOT NULL,
            C_CREDIT_LIM    DECIMAL(12,2) NOT NULL,
            C_MIDDLE        CHAR(2)       NOT NULL,
            C_CREDIT        CHAR(2)       NOT NULL,
            C_DISCOUNT      REAL          NOT NULL,
            C_DATA          VARCHAR(500)  NOT NULL,
            C_LAST          VARCHAR(16)   NOT NULL,
            C_FIRST         VARCHAR(16)   NOT NULL,
            C_STREET_1      VARCHAR(20)   NOT NULL,
            C_STREET_2      VARCHAR(20)   NOT NULL,
            C_CITY          VARCHAR(20)   NOT NULL,
            C_D_ID          SMALLINT      NOT NULL,
            C_W_ID          INTEGER       NOT NULL,
            C_DELIVERY_CNT  INTEGER       NOT NULL,
            C_BALANCE       DECIMAL(12,2) NOT NULL,
            C_YTD_PAYMENT   DECIMAL(12,2) NOT NULL,
            C_PAYMENT_CNT   INTEGER       NOT NULL
            )
            IN ts_customer_080
            INDEX IN is_customer_080
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 41081 ENDING AT 41600,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_DISTRICT.ddl

```
connect to TPCC in share mode;
DROP TABLE DISTRICT1;
CREATE TABLE DISTRICT1
            (
            D_NEXT_O_ID INTEGER       NOT NULL,
            D_TAX       REAL          NOT NULL,
            D_YTD       DECIMAL(12,2) NOT NULL,
            D_NAME      CHAR(10)      NOT NULL,
            D_STREET_1  CHAR(20)      NOT NULL,
            D_STREET_2  CHAR(20)      NOT NULL,
            D_CITY      CHAR(20)      NOT NULL,
            D_STATE     CHAR(2)       NOT NULL,
            D_ZIP       CHAR(9)       NOT NULL,
            D_ID        SMALLINT      NOT NULL,
```

```
 D_W_ID     INTEGER        NOT NULL
)
IN ts_dis_001
INDEX IN ts_dis_001
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 1 ENDING AT 520
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT2;
CREATE TABLE DISTRICT2
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_002
INDEX IN ts_dis_002
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 521 ENDING AT 1040
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT3;
CREATE TABLE DISTRICT3
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_003
INDEX IN ts_dis_003
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 1041 ENDING AT 1560
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT4;
CREATE TABLE DISTRICT4
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_004
INDEX IN ts_dis_004
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 1561 ENDING AT 2080
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT5;
CREATE TABLE DISTRICT5
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_005
INDEX IN ts_dis_005
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 2081 ENDING AT 2600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT6;
CREATE TABLE DISTRICT6
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_006
INDEX IN ts_dis_006
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 2601 ENDING AT 3120
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT7;
CREATE TABLE DISTRICT7
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_007
INDEX IN ts_dis_007
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 3121 ENDING AT 3640
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT8;
CREATE TABLE DISTRICT8
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_008
INDEX IN ts_dis_008
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 3641 ENDING AT 4160
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT9;
CREATE TABLE DISTRICT9
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_009
INDEX IN ts_dis_009
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 4161 ENDING AT 4680
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT10;
CREATE TABLE DISTRICT10
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
```

```
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_010
                INDEX IN ts_dis_010
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 4681 ENDING AT 5200
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT11;
CREATE TABLE DISTRICT11
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_011
                INDEX IN ts_dis_011
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 5201 ENDING AT 5720
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT12;
CREATE TABLE DISTRICT12
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_012
                INDEX IN ts_dis_012
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 5721 ENDING AT 6240
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT13;
CREATE TABLE DISTRICT13
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_013
                INDEX IN ts_dis_013
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 6241 ENDING AT 6760
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT14;
CREATE TABLE DISTRICT14
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_014
                INDEX IN ts_dis_014
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 6761 ENDING AT 7280
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT15;
CREATE TABLE DISTRICT15
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_015
                INDEX IN ts_dis_015
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 7281 ENDING AT 7800
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT16;
CREATE TABLE DISTRICT16
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_016
                INDEX IN ts_dis_016
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 7801 ENDING AT 8320
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT17;
CREATE TABLE DISTRICT17
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_017
                INDEX IN ts_dis_017
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 8321 ENDING AT 8840
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT18;
CREATE TABLE DISTRICT18
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_018
                INDEX IN ts_dis_018
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 8841 ENDING AT 9360
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT19;
```

```
CREATE TABLE DISTRICT19
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_019
                INDEX IN ts_dis_019
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 9361 ENDING AT 9880
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT20;
CREATE TABLE DISTRICT20
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_020
                INDEX IN ts_dis_020
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 9881 ENDING AT 10400
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT21;
CREATE TABLE DISTRICT21
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_021
                INDEX IN ts_dis_021
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 10401 ENDING AT 10920
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT22;
CREATE TABLE DISTRICT22
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_022
                INDEX IN ts_dis_022
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 10921 ENDING AT 11440
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT23;
CREATE TABLE DISTRICT23
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_023
                INDEX IN ts_dis_023
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 11441 ENDING AT 11960
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT24;
CREATE TABLE DISTRICT24
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_024
                INDEX IN ts_dis_024
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 11961 ENDING AT 12480
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT25;
CREATE TABLE DISTRICT25
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_025
                INDEX IN ts_dis_025
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 12481 ENDING AT 13000
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT26;
CREATE TABLE DISTRICT26
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_026
                INDEX IN ts_dis_026
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
                 D_W_ID STARTING FROM 13001 ENDING AT 13520
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT27;
CREATE TABLE DISTRICT27
                (
                 D_NEXT_O_ID INTEGER         NOT NULL,
                 D_TAX       REAL            NOT NULL,
                 D_YTD       DECIMAL(12,2)   NOT NULL,
                 D_NAME      CHAR(10)        NOT NULL,
                 D_STREET_1  CHAR(20)        NOT NULL,
                 D_STREET_2  CHAR(20)        NOT NULL,
                 D_CITY      CHAR(20)        NOT NULL,
                 D_STATE     CHAR(2)         NOT NULL,
                 D_ZIP       CHAR(9)         NOT NULL,
                 D_ID        SMALLINT        NOT NULL,
                 D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_027
                INDEX IN ts_dis_027
                ORGANIZE BY KEY SEQUENCE (
                 D_ID STARTING FROM 1 ENDING AT 10,
```

```
 D_W_ID STARTING FROM 13521 ENDING AT 14040
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT28;
CREATE TABLE DISTRICT28
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_028
INDEX IN ts_dis_028
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 14041 ENDING AT 14560
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT29;
CREATE TABLE DISTRICT29
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_029
INDEX IN ts_dis_029
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 14561 ENDING AT 15080
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT30;
CREATE TABLE DISTRICT30
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_030
INDEX IN ts_dis_030
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 15081 ENDING AT 15600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT31;
CREATE TABLE DISTRICT31
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_031
INDEX IN ts_dis_031
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 15601 ENDING AT 16120
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT32;
CREATE TABLE DISTRICT32
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_032
INDEX IN ts_dis_032
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 16121 ENDING AT 16640
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT33;
CREATE TABLE DISTRICT33
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_033
INDEX IN ts_dis_033
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 16641 ENDING AT 17160
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT34;
CREATE TABLE DISTRICT34
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_034
INDEX IN ts_dis_034
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 17161 ENDING AT 17680
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT35;
CREATE TABLE DISTRICT35
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_035
INDEX IN ts_dis_035
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 17681 ENDING AT 18200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT36;
CREATE TABLE DISTRICT36
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
```

```
 D_W_ID     INTEGER        NOT NULL
)
IN ts_dis_036
INDEX IN ts_dis_036
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 18201 ENDING AT 18720
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT37;
CREATE TABLE DISTRICT37
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_037
INDEX IN ts_dis_037
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 18721 ENDING AT 19240
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT38;
CREATE TABLE DISTRICT38
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_038
INDEX IN ts_dis_038
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 19241 ENDING AT 19760
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT39;
CREATE TABLE DISTRICT39
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_039
INDEX IN ts_dis_039
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 19761 ENDING AT 20280
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT40;
CREATE TABLE DISTRICT40
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_040
INDEX IN ts_dis_040
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 20281 ENDING AT 20800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT41;
CREATE TABLE DISTRICT41
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_041
INDEX IN ts_dis_041
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 20801 ENDING AT 21320
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT42;
CREATE TABLE DISTRICT42
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_042
INDEX IN ts_dis_042
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 21321 ENDING AT 21840
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT43;
CREATE TABLE DISTRICT43
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_043
INDEX IN ts_dis_043
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 21841 ENDING AT 22360
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT44;
CREATE TABLE DISTRICT44
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_044
INDEX IN ts_dis_044
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 22361 ENDING AT 22880
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT45;
CREATE TABLE DISTRICT45
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
```

```
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_045
                INDEX IN ts_dis_045
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 22881 ENDING AT 23400
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT46;
CREATE TABLE DISTRICT46
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_046
                INDEX IN ts_dis_046
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 23401 ENDING AT 23920
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT47;
CREATE TABLE DISTRICT47
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_047
                INDEX IN ts_dis_047
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 23921 ENDING AT 24440
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT48;
CREATE TABLE DISTRICT48
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_048
                INDEX IN ts_dis_048
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 24441 ENDING AT 24960
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT49;
CREATE TABLE DISTRICT49
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_049
                INDEX IN ts_dis_049
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 24961 ENDING AT 25480
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT50;
CREATE TABLE DISTRICT50
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_050
                INDEX IN ts_dis_050
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 25481 ENDING AT 26000
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT51;
CREATE TABLE DISTRICT51
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_051
                INDEX IN ts_dis_051
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 26001 ENDING AT 26520
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT52;
CREATE TABLE DISTRICT52
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_052
                INDEX IN ts_dis_052
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 26521 ENDING AT 27040
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT53;
CREATE TABLE DISTRICT53
                (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
                )
                IN ts_dis_053
                INDEX IN ts_dis_053
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 27041 ENDING AT 27560
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT54;
```

```
CREATE TABLE DISTRICT54
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_054
		INDEX IN ts_dis_054
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 27561 ENDING AT 28080
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT55;
CREATE TABLE DISTRICT55
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_055
		INDEX IN ts_dis_055
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 28081 ENDING AT 28600
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT56;
CREATE TABLE DISTRICT56
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_056
		INDEX IN ts_dis_056
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 28601 ENDING AT 29120
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT57;
CREATE TABLE DISTRICT57
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_057
		INDEX IN ts_dis_057
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 29121 ENDING AT 29640
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT58;
CREATE TABLE DISTRICT58
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_058
		INDEX IN ts_dis_058
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 29641 ENDING AT 30160
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT59;
CREATE TABLE DISTRICT59
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_059
		INDEX IN ts_dis_059
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 30161 ENDING AT 30680
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT60;
CREATE TABLE DISTRICT60
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_060
		INDEX IN ts_dis_060
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 30681 ENDING AT 31200
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT61;
CREATE TABLE DISTRICT61
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_061
		INDEX IN ts_dis_061
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
		 D_W_ID STARTING FROM 31201 ENDING AT 31720
		)
		ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT62;
CREATE TABLE DISTRICT62
		(
		 D_NEXT_O_ID INTEGER		NOT NULL,
		 D_TAX		REAL		NOT NULL,
		 D_YTD		DECIMAL(12,2)	NOT NULL,
		 D_NAME		CHAR(10)	NOT NULL,
		 D_STREET_1	CHAR(20)	NOT NULL,
		 D_STREET_2	CHAR(20)	NOT NULL,
		 D_CITY		CHAR(20)	NOT NULL,
		 D_STATE	CHAR(2)		NOT NULL,
		 D_ZIP		CHAR(9)		NOT NULL,
		 D_ID		SMALLINT	NOT NULL,
		 D_W_ID		INTEGER		NOT NULL
		)
		IN ts_dis_062
		INDEX IN ts_dis_062
		ORGANIZE BY KEY SEQUENCE (
		 D_ID STARTING FROM 1 ENDING AT 10,
```

```
                D_W_ID STARTING FROM 31721 ENDING AT 32240
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT63;
CREATE TABLE DISTRICT63
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_063
               INDEX IN ts_dis_063
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 32241 ENDING AT 32760
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT64;
CREATE TABLE DISTRICT64
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_064
               INDEX IN ts_dis_064
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 32761 ENDING AT 33280
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT65;
CREATE TABLE DISTRICT65
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_065
               INDEX IN ts_dis_065
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 33281 ENDING AT 33800
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT66;
CREATE TABLE DISTRICT66
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_066
               INDEX IN ts_dis_066
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 33801 ENDING AT 34320
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT67;
CREATE TABLE DISTRICT67
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_067
               INDEX IN ts_dis_067
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 34321 ENDING AT 34840
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT68;
CREATE TABLE DISTRICT68
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_068
               INDEX IN ts_dis_068
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 34841 ENDING AT 35360
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT69;
CREATE TABLE DISTRICT69
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_069
               INDEX IN ts_dis_069
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 35361 ENDING AT 35880
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT70;
CREATE TABLE DISTRICT70
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
                D_W_ID      INTEGER         NOT NULL
               )
               IN ts_dis_070
               INDEX IN ts_dis_070
               ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 35881 ENDING AT 36400
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT71;
CREATE TABLE DISTRICT71
               (
                D_NEXT_O_ID INTEGER         NOT NULL,
                D_TAX       REAL            NOT NULL,
                D_YTD       DECIMAL(12,2)   NOT NULL,
                D_NAME      CHAR(10)        NOT NULL,
                D_STREET_1  CHAR(20)        NOT NULL,
                D_STREET_2  CHAR(20)        NOT NULL,
                D_CITY      CHAR(20)        NOT NULL,
                D_STATE     CHAR(2)         NOT NULL,
                D_ZIP       CHAR(9)         NOT NULL,
                D_ID        SMALLINT        NOT NULL,
```

```
 D_W_ID     INTEGER        NOT NULL
)
IN ts_dis_071
INDEX IN ts_dis_071
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 36401 ENDING AT 36920
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT72;
CREATE TABLE DISTRICT72
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_072
INDEX IN ts_dis_072
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 36921 ENDING AT 37440
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT73;
CREATE TABLE DISTRICT73
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_073
INDEX IN ts_dis_073
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 37441 ENDING AT 37960
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT74;
CREATE TABLE DISTRICT74
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_074
INDEX IN ts_dis_074
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 37961 ENDING AT 38480
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT75;
CREATE TABLE DISTRICT75
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_075
INDEX IN ts_dis_075
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 38481 ENDING AT 39000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT76;
CREATE TABLE DISTRICT76
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_076
INDEX IN ts_dis_076
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 39001 ENDING AT 39520
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT77;
CREATE TABLE DISTRICT77
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_077
INDEX IN ts_dis_077
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 39521 ENDING AT 40040
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT78;
CREATE TABLE DISTRICT78
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_078
INDEX IN ts_dis_078
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 40041 ENDING AT 40560
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT79;
CREATE TABLE DISTRICT79
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
 D_STREET_1  CHAR(20)       NOT NULL,
 D_STREET_2  CHAR(20)       NOT NULL,
 D_CITY      CHAR(20)       NOT NULL,
 D_STATE     CHAR(2)        NOT NULL,
 D_ZIP       CHAR(9)        NOT NULL,
 D_ID        SMALLINT       NOT NULL,
 D_W_ID      INTEGER        NOT NULL
)
IN ts_dis_079
INDEX IN ts_dis_079
ORGANIZE BY KEY SEQUENCE (
 D_ID STARTING FROM 1 ENDING AT 10,
 D_W_ID STARTING FROM 40561 ENDING AT 41080
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT80;
CREATE TABLE DISTRICT80
(
 D_NEXT_O_ID INTEGER        NOT NULL,
 D_TAX       REAL           NOT NULL,
 D_YTD       DECIMAL(12,2)  NOT NULL,
 D_NAME      CHAR(10)       NOT NULL,
```

```
                D_STREET_1  CHAR(20)       NOT NULL,
                D_STREET_2  CHAR(20)       NOT NULL,
                D_CITY      CHAR(20)       NOT NULL,
                D_STATE     CHAR(2)        NOT NULL,
                D_ZIP       CHAR(9)        NOT NULL,
                D_ID        SMALLINT       NOT NULL,
                D_W_ID      INTEGER        NOT NULL
                )
                IN ts_dis_080
                INDEX IN ts_dis_080
                ORGANIZE BY KEY SEQUENCE (
                D_ID STARTING FROM 1 ENDING AT 10,
                D_W_ID STARTING FROM 41081 ENDING AT 41600
                )
                ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_HISTORY.ddl

```
connect to TPCC in share mode; connect to TPCC in share mode;
DROP TABLE HISTORY1;
CREATE TABLE HISTORY1
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_001
                INDEX IN ts_history_001;
ALTER TABLE HISTORY1 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY2;
CREATE TABLE HISTORY2
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_002
                INDEX IN ts_history_002;
ALTER TABLE HISTORY2 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY3;
CREATE TABLE HISTORY3
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_003
                INDEX IN ts_history_003;
ALTER TABLE HISTORY3 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY4;
CREATE TABLE HISTORY4
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_004
                INDEX IN ts_history_004;
ALTER TABLE HISTORY4 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY5;
CREATE TABLE HISTORY5
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_005
                INDEX IN ts_history_005;
ALTER TABLE HISTORY5 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY6;
CREATE TABLE HISTORY6
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_006
                INDEX IN ts_history_006;
ALTER TABLE HISTORY6 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY7;
CREATE TABLE HISTORY7
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_007
                INDEX IN ts_history_007;
ALTER TABLE HISTORY7 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY8;
CREATE TABLE HISTORY8
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_008
                INDEX IN ts_history_008;
ALTER TABLE HISTORY8 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY9;
CREATE TABLE HISTORY9
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_009
                INDEX IN ts_history_009;
ALTER TABLE HISTORY9 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY10;
CREATE TABLE HISTORY10
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_010
                INDEX IN ts_history_010;
ALTER TABLE HISTORY10 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY11;
CREATE TABLE HISTORY11
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_011
                INDEX IN ts_history_011;
ALTER TABLE HISTORY11 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY12;
CREATE TABLE HISTORY12
                (
```

```
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_012
                INDEX IN ts_history_012;
ALTER TABLE HISTORY12 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY13;
CREATE TABLE HISTORY13
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_013
                INDEX IN ts_history_013;
ALTER TABLE HISTORY13 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY14;
CREATE TABLE HISTORY14
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_014
                INDEX IN ts_history_014;
ALTER TABLE HISTORY14 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY15;
CREATE TABLE HISTORY15
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_015
                INDEX IN ts_history_015;
ALTER TABLE HISTORY15 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY16;
CREATE TABLE HISTORY16
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_016
                INDEX IN ts_history_016;
ALTER TABLE HISTORY16 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY17;
CREATE TABLE HISTORY17
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_017
                INDEX IN ts_history_017;
ALTER TABLE HISTORY17 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY18;
CREATE TABLE HISTORY18
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_018
                INDEX IN ts_history_018;
ALTER TABLE HISTORY18 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY19;
CREATE TABLE HISTORY19
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_019
                INDEX IN ts_history_019;
ALTER TABLE HISTORY19 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY20;
CREATE TABLE HISTORY20
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_020
                INDEX IN ts_history_020;
ALTER TABLE HISTORY20 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY21;
CREATE TABLE HISTORY21
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_021
                INDEX IN ts_history_021;
ALTER TABLE HISTORY21 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY22;
CREATE TABLE HISTORY22
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_022
                INDEX IN ts_history_022;
ALTER TABLE HISTORY22 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY23;
CREATE TABLE HISTORY23
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_023
                INDEX IN ts_history_023;
ALTER TABLE HISTORY23 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY24;
CREATE TABLE HISTORY24
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
```

```
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_024
                INDEX IN ts_history_024;
ALTER TABLE HISTORY24 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY25;
CREATE TABLE HISTORY25
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_025
                INDEX IN ts_history_025;
ALTER TABLE HISTORY25 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY26;
CREATE TABLE HISTORY26
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_026
                INDEX IN ts_history_026;
ALTER TABLE HISTORY26 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY27;
CREATE TABLE HISTORY27
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_027
                INDEX IN ts_history_027;
ALTER TABLE HISTORY27 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY28;
CREATE TABLE HISTORY28
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_028
                INDEX IN ts_history_028;
ALTER TABLE HISTORY28 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY29;
CREATE TABLE HISTORY29
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_029
                INDEX IN ts_history_029;
ALTER TABLE HISTORY29 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY30;
CREATE TABLE HISTORY30
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_030
                INDEX IN ts_history_030;
ALTER TABLE HISTORY30 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY31;
CREATE TABLE HISTORY31
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_031
                INDEX IN ts_history_031;
ALTER TABLE HISTORY31 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY32;
CREATE TABLE HISTORY32
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_032
                INDEX IN ts_history_032;
ALTER TABLE HISTORY32 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY33;
CREATE TABLE HISTORY33
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_033
                INDEX IN ts_history_033;
ALTER TABLE HISTORY33 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY34;
CREATE TABLE HISTORY34
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_034
                INDEX IN ts_history_034;
ALTER TABLE HISTORY34 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY35;
CREATE TABLE HISTORY35
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_035
                INDEX IN ts_history_035;
ALTER TABLE HISTORY35 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY36;
CREATE TABLE HISTORY36
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_036
                INDEX IN ts_history_036;
ALTER TABLE HISTORY36 APPEND ON;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY37;
CREATE TABLE HISTORY37
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_037
        INDEX IN ts_history_037;
ALTER TABLE HISTORY37 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY38;
CREATE TABLE HISTORY38
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_038
        INDEX IN ts_history_038;
ALTER TABLE HISTORY38 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY39;
CREATE TABLE HISTORY39
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_039
        INDEX IN ts_history_039;
ALTER TABLE HISTORY39 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY40;
CREATE TABLE HISTORY40
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_040
        INDEX IN ts_history_040;
ALTER TABLE HISTORY40 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY41;
CREATE TABLE HISTORY41
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_041
        INDEX IN ts_history_041;
ALTER TABLE HISTORY41 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY42;
CREATE TABLE HISTORY42
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_042
        INDEX IN ts_history_042;
ALTER TABLE HISTORY42 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY43;
CREATE TABLE HISTORY43
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_043
        INDEX IN ts_history_043;
ALTER TABLE HISTORY43 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY44;
CREATE TABLE HISTORY44
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_044
        INDEX IN ts_history_044;
ALTER TABLE HISTORY44 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY45;
CREATE TABLE HISTORY45
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_045
        INDEX IN ts_history_045;
ALTER TABLE HISTORY45 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY46;
CREATE TABLE HISTORY46
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_046
        INDEX IN ts_history_046;
ALTER TABLE HISTORY46 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY47;
CREATE TABLE HISTORY47
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_047
        INDEX IN ts_history_047;
ALTER TABLE HISTORY47 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY48;
CREATE TABLE HISTORY48
        (
         H_C_ID         INTEGER      NOT NULL,
         H_C_D_ID       SMALLINT     NOT NULL,
         H_C_W_ID       INTEGER      NOT NULL,
         H_D_ID         SMALLINT     NOT NULL,
         H_W_ID         INTEGER      NOT NULL,
         H_DATE         TIMESTAMP    NOT NULL,
         H_AMOUNT       DECIMAL(6,2) NOT NULL,
         H_DATA         CHAR(24)     NOT NULL
        )
        IN ts_history_048
        INDEX IN ts_history_048;
ALTER TABLE HISTORY48 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY49;
CREATE TABLE HISTORY49
        (
         H_C_ID         INTEGER      NOT NULL,
```

```
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_049
                INDEX IN ts_history_049;
ALTER TABLE HISTORY49 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY50;
CREATE TABLE HISTORY50
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_050
                INDEX IN ts_history_050;
ALTER TABLE HISTORY50 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY51;
CREATE TABLE HISTORY51
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_051
                INDEX IN ts_history_051;
ALTER TABLE HISTORY51 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY52;
CREATE TABLE HISTORY52
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_052
                INDEX IN ts_history_052;
ALTER TABLE HISTORY52 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY53;
CREATE TABLE HISTORY53
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_053
                INDEX IN ts_history_053;
ALTER TABLE HISTORY53 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY54;
CREATE TABLE HISTORY54
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_054
                INDEX IN ts_history_054;
ALTER TABLE HISTORY54 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY55;
CREATE TABLE HISTORY55
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_055
                INDEX IN ts_history_055;
ALTER TABLE HISTORY55 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY56;
CREATE TABLE HISTORY56
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_056
                INDEX IN ts_history_056;
ALTER TABLE HISTORY56 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY57;
CREATE TABLE HISTORY57
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_057
                INDEX IN ts_history_057;
ALTER TABLE HISTORY57 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY58;
CREATE TABLE HISTORY58
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_058
                INDEX IN ts_history_058;
ALTER TABLE HISTORY58 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY59;
CREATE TABLE HISTORY59
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_059
                INDEX IN ts_history_059;
ALTER TABLE HISTORY59 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY60;
CREATE TABLE HISTORY60
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
                H_DATA          CHAR(24)     NOT NULL
                )
                IN ts_history_060
                INDEX IN ts_history_060;
ALTER TABLE HISTORY60 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY61;
CREATE TABLE HISTORY61
                (
                H_C_ID          INTEGER      NOT NULL,
                H_C_D_ID        SMALLINT     NOT NULL,
                H_C_W_ID        INTEGER      NOT NULL,
                H_D_ID          SMALLINT     NOT NULL,
                H_W_ID          INTEGER      NOT NULL,
                H_DATE          TIMESTAMP    NOT NULL,
                H_AMOUNT        DECIMAL(6,2) NOT NULL,
```

```
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_061
    INDEX IN ts_history_061;
ALTER TABLE HISTORY61 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY62;
CREATE TABLE HISTORY62
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_062
    INDEX IN ts_history_062;
ALTER TABLE HISTORY62 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY63;
CREATE TABLE HISTORY63
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_063
    INDEX IN ts_history_063;
ALTER TABLE HISTORY63 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY64;
CREATE TABLE HISTORY64
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_064
    INDEX IN ts_history_064;
ALTER TABLE HISTORY64 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY65;
CREATE TABLE HISTORY65
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_065
    INDEX IN ts_history_065;
ALTER TABLE HISTORY65 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY66;
CREATE TABLE HISTORY66
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_066
    INDEX IN ts_history_066;
ALTER TABLE HISTORY66 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY67;
CREATE TABLE HISTORY67
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_067
    INDEX IN ts_history_067;
ALTER TABLE HISTORY67 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY68;
CREATE TABLE HISTORY68
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_068
    INDEX IN ts_history_068;
ALTER TABLE HISTORY68 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY69;
CREATE TABLE HISTORY69
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_069
    INDEX IN ts_history_069;
ALTER TABLE HISTORY69 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY70;
CREATE TABLE HISTORY70
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_070
    INDEX IN ts_history_070;
ALTER TABLE HISTORY70 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY71;
CREATE TABLE HISTORY71
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_071
    INDEX IN ts_history_071;
ALTER TABLE HISTORY71 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY72;
CREATE TABLE HISTORY72
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_072
    INDEX IN ts_history_072;
ALTER TABLE HISTORY72 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY73;
CREATE TABLE HISTORY73
    (
    H_C_ID          INTEGER      NOT NULL,
    H_C_D_ID        SMALLINT     NOT NULL,
    H_C_W_ID        INTEGER      NOT NULL,
    H_D_ID          SMALLINT     NOT NULL,
    H_W_ID          INTEGER      NOT NULL,
    H_DATE          TIMESTAMP    NOT NULL,
    H_AMOUNT        DECIMAL(6,2) NOT NULL,
    H_DATA          CHAR(24)     NOT NULL
    )
    IN ts_history_073
    INDEX IN ts_history_073;
ALTER TABLE HISTORY73 APPEND ON;
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE HISTORY74;
CREATE TABLE HISTORY74
              (
               H_C_ID         INTEGER      NOT NULL,
               H_C_D_ID       SMALLINT     NOT NULL,
               H_C_W_ID       INTEGER      NOT NULL,
               H_D_ID         SMALLINT     NOT NULL,
               H_W_ID         INTEGER      NOT NULL,
               H_DATE         TIMESTAMP    NOT NULL,
               H_AMOUNT       DECIMAL(6,2) NOT NULL,
               H_DATA         CHAR(24)     NOT NULL
              )
              IN ts_history_074
              INDEX IN ts_history_074;
ALTER TABLE HISTORY74 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY75;
CREATE TABLE HISTORY75
              (
               H_C_ID         INTEGER      NOT NULL,
               H_C_D_ID       SMALLINT     NOT NULL,
               H_C_W_ID       INTEGER      NOT NULL,
               H_D_ID         SMALLINT     NOT NULL,
               H_W_ID         INTEGER      NOT NULL,
               H_DATE         TIMESTAMP    NOT NULL,
               H_AMOUNT       DECIMAL(6,2) NOT NULL,
               H_DATA         CHAR(24)     NOT NULL
              )
              IN ts_history_075
              INDEX IN ts_history_075;
ALTER TABLE HISTORY75 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY76;
CREATE TABLE HISTORY76
              (
               H_C_ID         INTEGER      NOT NULL,
               H_C_D_ID       SMALLINT     NOT NULL,
               H_C_W_ID       INTEGER      NOT NULL,
               H_D_ID         SMALLINT     NOT NULL,
               H_W_ID         INTEGER      NOT NULL,
               H_DATE         TIMESTAMP    NOT NULL,
               H_AMOUNT       DECIMAL(6,2) NOT NULL,
               H_DATA         CHAR(24)     NOT NULL
              )
              IN ts_history_076
              INDEX IN ts_history_076;
ALTER TABLE HISTORY76 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY77;
CREATE TABLE HISTORY77
              (
               H_C_ID         INTEGER      NOT NULL,
               H_C_D_ID       SMALLINT     NOT NULL,
               H_C_W_ID       INTEGER      NOT NULL,
               H_D_ID         SMALLINT     NOT NULL,
               H_W_ID         INTEGER      NOT NULL,
               H_DATE         TIMESTAMP    NOT NULL,
               H_AMOUNT       DECIMAL(6,2) NOT NULL,
               H_DATA         CHAR(24)     NOT NULL
              )
              IN ts_history_077
              INDEX IN ts_history_077;
ALTER TABLE HISTORY77 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY78;
CREATE TABLE HISTORY78
              (
               H_C_ID         INTEGER      NOT NULL,
               H_C_D_ID       SMALLINT     NOT NULL,
               H_C_W_ID       INTEGER      NOT NULL,
               H_D_ID         SMALLINT     NOT NULL,
               H_W_ID         INTEGER      NOT NULL,
               H_DATE         TIMESTAMP    NOT NULL,
               H_AMOUNT       DECIMAL(6,2) NOT NULL,
               H_DATA         CHAR(24)     NOT NULL
              )
              IN ts_history_078
              INDEX IN ts_history_078;
ALTER TABLE HISTORY78 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY79;
CREATE TABLE HISTORY79
              (
               H_C_ID         INTEGER      NOT NULL,
               H_C_D_ID       SMALLINT     NOT NULL,
               H_C_W_ID       INTEGER      NOT NULL,
               H_D_ID         SMALLINT     NOT NULL,
               H_W_ID         INTEGER      NOT NULL,
               H_DATE         TIMESTAMP    NOT NULL,
               H_AMOUNT       DECIMAL(6,2) NOT NULL,
               H_DATA         CHAR(24)     NOT NULL
              )
              IN ts_history_079
              INDEX IN ts_history_079;
ALTER TABLE HISTORY79 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY80;
CREATE TABLE HISTORY80
              (
               H_C_ID         INTEGER      NOT NULL,
               H_C_D_ID       SMALLINT     NOT NULL,
               H_C_W_ID       INTEGER      NOT NULL,
               H_D_ID         SMALLINT     NOT NULL,
               H_W_ID         INTEGER      NOT NULL,
               H_DATE         TIMESTAMP    NOT NULL,
               H_AMOUNT       DECIMAL(6,2) NOT NULL,
               H_DATA         CHAR(24)     NOT NULL
              )
              IN ts_history_080
              INDEX IN ts_history_080;
ALTER TABLE HISTORY80 APPEND ON;
connect reset;
```

DDL/CRTB_ITEM.ddl

```
connect to TPCC in share mode;
DROP TABLE ITEM;
CREATE TABLE ITEM
              (
               I_NAME         CHAR(24)     NOT NULL,
               I_PRICE        DECIMAL(5,2) NOT NULL,
               I_DATA         VARCHAR(50)  NOT NULL,
               I_IM_ID        INTEGER      NOT NULL,
               I_ID           INTEGER      NOT NULL
              )
              IN ts_item
              INDEX IN ts_item
              ORGANIZE BY KEY SEQUENCE (
               I_ID STARTING FROM 1 ENDING AT 100000
              )
              ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;
connect reset;
```

DDL/CRTB_NEW_ORDERA.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA1;
CREATE TABLE NEW_ORDERA1
              (
               NO_O_ID        INTEGER    NOT NULL,
               NO_D_ID        SMALLINT   NOT NULL,
               NO_W_ID        INTEGER    NOT NULL
              )
              IN ts_newordA_001
              INDEX IN ts_newordA_001
              ORGANIZE BY KEY SEQUENCE (
               NO_W_ID STARTING FROM 1 ENDING AT 520,
               NO_D_ID STARTING FROM 1 ENDING AT 10,
               NO_O_ID STARTING FROM 1900 ENDING AT 3675
              )
              ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA2;
CREATE TABLE NEW_ORDERA2
              (
               NO_O_ID        INTEGER    NOT NULL,
               NO_D_ID        SMALLINT   NOT NULL,
               NO_W_ID        INTEGER    NOT NULL
              )
              IN ts_newordA_002
              INDEX IN ts_newordA_002
              ORGANIZE BY KEY SEQUENCE (
               NO_W_ID STARTING FROM 521 ENDING AT 1040,
               NO_D_ID STARTING FROM 1 ENDING AT 10,
               NO_O_ID STARTING FROM 1900 ENDING AT 3675
              )
              ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA3;
CREATE TABLE NEW_ORDERA3
              (
               NO_O_ID        INTEGER    NOT NULL,
               NO_D_ID        SMALLINT   NOT NULL,
               NO_W_ID        INTEGER    NOT NULL
              )
              IN ts_newordA_003
              INDEX IN ts_newordA_003
              ORGANIZE BY KEY SEQUENCE (
               NO_W_ID STARTING FROM 1041 ENDING AT 1560,
               NO_D_ID STARTING FROM 1 ENDING AT 10,
               NO_O_ID STARTING FROM 1900 ENDING AT 3675
              )
              ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA4;
CREATE TABLE NEW_ORDERA4
              (
               NO_O_ID        INTEGER    NOT NULL,
               NO_D_ID        SMALLINT   NOT NULL,
               NO_W_ID        INTEGER    NOT NULL
              )
              IN ts_newordA_004
              INDEX IN ts_newordA_004
              ORGANIZE BY KEY SEQUENCE (
               NO_W_ID STARTING FROM 1561 ENDING AT 2080,
               NO_D_ID STARTING FROM 1 ENDING AT 10,
               NO_O_ID STARTING FROM 1900 ENDING AT 3675
              )
```

```
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA5;
CREATE TABLE NEW_ORDERA5
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_005
                INDEX IN ts_newordA_005
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 2081 ENDING AT 2600,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA6;
CREATE TABLE NEW_ORDERA6
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_006
                INDEX IN ts_newordA_006
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 2601 ENDING AT 3120,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA7;
CREATE TABLE NEW_ORDERA7
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_007
                INDEX IN ts_newordA_007
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 3121 ENDING AT 3640,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA8;
CREATE TABLE NEW_ORDERA8
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_008
                INDEX IN ts_newordA_008
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 3641 ENDING AT 4160,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA9;
CREATE TABLE NEW_ORDERA9
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_009
                INDEX IN ts_newordA_009
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 4161 ENDING AT 4680,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA10;
CREATE TABLE NEW_ORDERA10
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_010
                INDEX IN ts_newordA_010
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 4681 ENDING AT 5200,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA11;
CREATE TABLE NEW_ORDERA11
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_011
                INDEX IN ts_newordA_011
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 5201 ENDING AT 5720,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA12;
CREATE TABLE NEW_ORDERA12
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_012
                INDEX IN ts_newordA_012
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 5721 ENDING AT 6240,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA13;
CREATE TABLE NEW_ORDERA13
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_013
                INDEX IN ts_newordA_013
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 6241 ENDING AT 6760,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA14;
CREATE TABLE NEW_ORDERA14
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_014
                INDEX IN ts_newordA_014
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 6761 ENDING AT 7280,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA15;
CREATE TABLE NEW_ORDERA15
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_015
                INDEX IN ts_newordA_015
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 7281 ENDING AT 7800,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA16;
CREATE TABLE NEW_ORDERA16
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_016
                INDEX IN ts_newordA_016
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 7801 ENDING AT 8320,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA17;
CREATE TABLE NEW_ORDERA17
                (
```

```
                NO_O_ID        INTEGER    NOT NULL,
                NO_D_ID        SMALLINT   NOT NULL,
                NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_017
                INDEX IN ts_newordA_017
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 8321 ENDING AT 8840,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA18;
CREATE TABLE NEW_ORDERA18
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_018
                INDEX IN ts_newordA_018
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 8841 ENDING AT 9360,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA19;
CREATE TABLE NEW_ORDERA19
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_019
                INDEX IN ts_newordA_019
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 9361 ENDING AT 9880,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA20;
CREATE TABLE NEW_ORDERA20
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_020
                INDEX IN ts_newordA_020
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 9881 ENDING AT 10400,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA21;
CREATE TABLE NEW_ORDERA21
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_021
                INDEX IN ts_newordA_021
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 10401 ENDING AT 10920,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA22;
CREATE TABLE NEW_ORDERA22
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_022
                INDEX IN ts_newordA_022
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 10921 ENDING AT 11440,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA23;
CREATE TABLE NEW_ORDERA23
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_023
                INDEX IN ts_newordA_023
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 11441 ENDING AT 11960,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA24;
CREATE TABLE NEW_ORDERA24
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_024
                INDEX IN ts_newordA_024
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 11961 ENDING AT 12480,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA25;
CREATE TABLE NEW_ORDERA25
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_025
                INDEX IN ts_newordA_025
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 12481 ENDING AT 13000,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA26;
CREATE TABLE NEW_ORDERA26
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_026
                INDEX IN ts_newordA_026
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 13001 ENDING AT 13520,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA27;
CREATE TABLE NEW_ORDERA27
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_027
                INDEX IN ts_newordA_027
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 13521 ENDING AT 14040,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA28;
CREATE TABLE NEW_ORDERA28
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_028
                INDEX IN ts_newordA_028
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 14041 ENDING AT 14560,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA29;
CREATE TABLE NEW_ORDERA29
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordA_029
                INDEX IN ts_newordA_029
```

```
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 14561 ENDING AT 15080,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA30;
CREATE TABLE NEW_ORDERA30
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_030
                INDEX IN ts_newordA_030
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 15081 ENDING AT 15600,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA31;
CREATE TABLE NEW_ORDERA31
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_031
                INDEX IN ts_newordA_031
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 15601 ENDING AT 16120,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA32;
CREATE TABLE NEW_ORDERA32
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_032
                INDEX IN ts_newordA_032
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 16121 ENDING AT 16640,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA33;
CREATE TABLE NEW_ORDERA33
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_033
                INDEX IN ts_newordA_033
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 16641 ENDING AT 17160,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA34;
CREATE TABLE NEW_ORDERA34
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_034
                INDEX IN ts_newordA_034
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 17161 ENDING AT 17680,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA35;
CREATE TABLE NEW_ORDERA35
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_035
                INDEX IN ts_newordA_035
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 17681 ENDING AT 18200,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA36;
CREATE TABLE NEW_ORDERA36
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_036
                INDEX IN ts_newordA_036
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 18201 ENDING AT 18720,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA37;
CREATE TABLE NEW_ORDERA37
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_037
                INDEX IN ts_newordA_037
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 18721 ENDING AT 19240,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA38;
CREATE TABLE NEW_ORDERA38
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_038
                INDEX IN ts_newordA_038
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 19241 ENDING AT 19760,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA39;
CREATE TABLE NEW_ORDERA39
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_039
                INDEX IN ts_newordA_039
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 19761 ENDING AT 20280,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA40;
CREATE TABLE NEW_ORDERA40
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_040
                INDEX IN ts_newordA_040
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 20281 ENDING AT 20800,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA41;
CREATE TABLE NEW_ORDERA41
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_041
                INDEX IN ts_newordA_041
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 20801 ENDING AT 21320,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA42;
CREATE TABLE NEW_ORDERA42
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_042
                INDEX IN ts_newordA_042
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 21321 ENDING AT 21840,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA43;
CREATE TABLE NEW_ORDERA43
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_043
                INDEX IN ts_newordA_043
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 21841 ENDING AT 22360,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA44;
CREATE TABLE NEW_ORDERA44
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_044
                INDEX IN ts_newordA_044
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 22361 ENDING AT 22880,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA45;
CREATE TABLE NEW_ORDERA45
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_045
                INDEX IN ts_newordA_045
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 22881 ENDING AT 23400,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA46;
CREATE TABLE NEW_ORDERA46
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_046
                INDEX IN ts_newordA_046
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 23401 ENDING AT 23920,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA47;
CREATE TABLE NEW_ORDERA47
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_047
                INDEX IN ts_newordA_047
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 23921 ENDING AT 24440,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA48;
CREATE TABLE NEW_ORDERA48
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_048
                INDEX IN ts_newordA_048
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 24441 ENDING AT 24960,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA49;
CREATE TABLE NEW_ORDERA49
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_049
                INDEX IN ts_newordA_049
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 24961 ENDING AT 25480,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA50;
CREATE TABLE NEW_ORDERA50
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_050
                INDEX IN ts_newordA_050
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 25481 ENDING AT 26000,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA51;
CREATE TABLE NEW_ORDERA51
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_051
                INDEX IN ts_newordA_051
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 26001 ENDING AT 26520,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA52;
CREATE TABLE NEW_ORDERA52
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_052
                INDEX IN ts_newordA_052
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 26521 ENDING AT 27040,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA53;
CREATE TABLE NEW_ORDERA53
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_053
                INDEX IN ts_newordA_053
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 27041 ENDING AT 27560,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA54;
CREATE TABLE NEW_ORDERA54
                (
                 NO_O_ID        INTEGER    NOT NULL,
```

```
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_054
               INDEX IN ts_newordA_054
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 27561 ENDING AT 28080,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA55;
CREATE TABLE NEW_ORDERA55
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_055
               INDEX IN ts_newordA_055
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 28081 ENDING AT 28600,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA56;
CREATE TABLE NEW_ORDERA56
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_056
               INDEX IN ts_newordA_056
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 28601 ENDING AT 29120,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA57;
CREATE TABLE NEW_ORDERA57
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_057
               INDEX IN ts_newordA_057
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 29121 ENDING AT 29640,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA58;
CREATE TABLE NEW_ORDERA58
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_058
               INDEX IN ts_newordA_058
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 29641 ENDING AT 30160,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA59;
CREATE TABLE NEW_ORDERA59
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_059
               INDEX IN ts_newordA_059
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 30161 ENDING AT 30680,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA60;
CREATE TABLE NEW_ORDERA60
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_060
               INDEX IN ts_newordA_060
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 30681 ENDING AT 31200,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA61;
CREATE TABLE NEW_ORDERA61
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_061
               INDEX IN ts_newordA_061
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 31201 ENDING AT 31720,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA62;
CREATE TABLE NEW_ORDERA62
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_062
               INDEX IN ts_newordA_062
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 31721 ENDING AT 32240,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA63;
CREATE TABLE NEW_ORDERA63
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_063
               INDEX IN ts_newordA_063
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 32241 ENDING AT 32760,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA64;
CREATE TABLE NEW_ORDERA64
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_064
               INDEX IN ts_newordA_064
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 32761 ENDING AT 33280,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA65;
CREATE TABLE NEW_ORDERA65
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_065
               INDEX IN ts_newordA_065
               ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 33281 ENDING AT 33800,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA66;
CREATE TABLE NEW_ORDERA66
               (
                NO_O_ID        INTEGER     NOT NULL,
                NO_D_ID        SMALLINT    NOT NULL,
                NO_W_ID        INTEGER     NOT NULL
               )
               IN ts_newordA_066
               INDEX IN ts_newordA_066
               ORGANIZE BY KEY SEQUENCE (
```

```
                NO_W_ID STARTING FROM 33801 ENDING AT 34320,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA67;
CREATE TABLE NEW_ORDERA67
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_067
                INDEX IN ts_newordA_067
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 34321 ENDING AT 34840,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA68;
CREATE TABLE NEW_ORDERA68
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_068
                INDEX IN ts_newordA_068
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 34841 ENDING AT 35360,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA69;
CREATE TABLE NEW_ORDERA69
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_069
                INDEX IN ts_newordA_069
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 35361 ENDING AT 35880,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA70;
CREATE TABLE NEW_ORDERA70
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_070
                INDEX IN ts_newordA_070
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 35881 ENDING AT 36400,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA71;
CREATE TABLE NEW_ORDERA71
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_071
                INDEX IN ts_newordA_071
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 36401 ENDING AT 36920,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA72;
CREATE TABLE NEW_ORDERA72
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_072
                INDEX IN ts_newordA_072
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 36921 ENDING AT 37440,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA73;
CREATE TABLE NEW_ORDERA73
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_073
                INDEX IN ts_newordA_073
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 37441 ENDING AT 37960,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA74;
CREATE TABLE NEW_ORDERA74
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_074
                INDEX IN ts_newordA_074
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 37961 ENDING AT 38480,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA75;
CREATE TABLE NEW_ORDERA75
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_075
                INDEX IN ts_newordA_075
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 38481 ENDING AT 39000,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA76;
CREATE TABLE NEW_ORDERA76
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_076
                INDEX IN ts_newordA_076
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 39001 ENDING AT 39520,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA77;
CREATE TABLE NEW_ORDERA77
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_077
                INDEX IN ts_newordA_077
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 39521 ENDING AT 40040,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA78;
CREATE TABLE NEW_ORDERA78
                (
                NO_O_ID         INTEGER     NOT NULL,
                NO_D_ID         SMALLINT    NOT NULL,
                NO_W_ID         INTEGER     NOT NULL
                )
                IN ts_newordA_078
                INDEX IN ts_newordA_078
                ORGANIZE BY KEY SEQUENCE (
                NO_W_ID STARTING FROM 40041 ENDING AT 40560,
                NO_D_ID STARTING FROM 1 ENDING AT 10,
                NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA79;
CREATE TABLE NEW_ORDERA79
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_079
                INDEX IN ts_newordA_079
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 40561 ENDING AT 41080,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA80;
CREATE TABLE NEW_ORDERA80
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordA_080
                INDEX IN ts_newordA_080
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 41081 ENDING AT 41600,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_NEW_ORDERB.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB1;
CREATE TABLE NEW_ORDERB1
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_001
                INDEX IN ts_newordB_001
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 1 ENDING AT 520,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB2;
CREATE TABLE NEW_ORDERB2
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_002
                INDEX IN ts_newordB_002
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 521 ENDING AT 1040,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB3;
CREATE TABLE NEW_ORDERB3
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_003
                INDEX IN ts_newordB_003
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 1041 ENDING AT 1560,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB4;
CREATE TABLE NEW_ORDERB4
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_004
                INDEX IN ts_newordB_004
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 1561 ENDING AT 2080,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB5;
CREATE TABLE NEW_ORDERB5
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_005
                INDEX IN ts_newordB_005
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 2081 ENDING AT 2600,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB6;
CREATE TABLE NEW_ORDERB6
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_006
                INDEX IN ts_newordB_006
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 2601 ENDING AT 3120,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB7;
CREATE TABLE NEW_ORDERB7
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_007
                INDEX IN ts_newordB_007
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 3121 ENDING AT 3640,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB8;
CREATE TABLE NEW_ORDERB8
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_008
                INDEX IN ts_newordB_008
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 3641 ENDING AT 4160,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB9;
CREATE TABLE NEW_ORDERB9
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_009
                INDEX IN ts_newordB_009
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 4161 ENDING AT 4680,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB10;
CREATE TABLE NEW_ORDERB10
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_010
                INDEX IN ts_newordB_010
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 4681 ENDING AT 5200,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB11;
CREATE TABLE NEW_ORDERB11
```

```
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_011
                INDEX IN ts_newordB_011
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 5201 ENDING AT 5720,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB12;
CREATE TABLE NEW_ORDERB12
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_012
                INDEX IN ts_newordB_012
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 5721 ENDING AT 6240,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB13;
CREATE TABLE NEW_ORDERB13
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_013
                INDEX IN ts_newordB_013
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 6241 ENDING AT 6760,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB14;
CREATE TABLE NEW_ORDERB14
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_014
                INDEX IN ts_newordB_014
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 6761 ENDING AT 7280,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB15;
CREATE TABLE NEW_ORDERB15
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_015
                INDEX IN ts_newordB_015
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 7281 ENDING AT 7800,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB16;
CREATE TABLE NEW_ORDERB16
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_016
                INDEX IN ts_newordB_016
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 7801 ENDING AT 8320,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB17;
CREATE TABLE NEW_ORDERB17
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_017
                INDEX IN ts_newordB_017
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 8321 ENDING AT 8840,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB18;
CREATE TABLE NEW_ORDERB18
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_018
                INDEX IN ts_newordB_018
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 8841 ENDING AT 9360,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB19;
CREATE TABLE NEW_ORDERB19
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_019
                INDEX IN ts_newordB_019
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 9361 ENDING AT 9880,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB20;
CREATE TABLE NEW_ORDERB20
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_020
                INDEX IN ts_newordB_020
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 9881 ENDING AT 10400,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB21;
CREATE TABLE NEW_ORDERB21
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_021
                INDEX IN ts_newordB_021
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 10401 ENDING AT 10920,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB22;
CREATE TABLE NEW_ORDERB22
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_022
                INDEX IN ts_newordB_022
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 10921 ENDING AT 11440,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB23;
CREATE TABLE NEW_ORDERB23
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_023
```

```
                INDEX IN ts_newordB_023
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 11441 ENDING AT 11960,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB24;
CREATE TABLE NEW_ORDERB24
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_024
                INDEX IN ts_newordB_024
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 11961 ENDING AT 12480,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB25;
CREATE TABLE NEW_ORDERB25
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_025
                INDEX IN ts_newordB_025
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 12481 ENDING AT 13000,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB26;
CREATE TABLE NEW_ORDERB26
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_026
                INDEX IN ts_newordB_026
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 13001 ENDING AT 13520,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB27;
CREATE TABLE NEW_ORDERB27
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_027
                INDEX IN ts_newordB_027
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 13521 ENDING AT 14040,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB28;
CREATE TABLE NEW_ORDERB28
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_028
                INDEX IN ts_newordB_028
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 14041 ENDING AT 14560,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB29;
CREATE TABLE NEW_ORDERB29
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_029
                INDEX IN ts_newordB_029
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 14561 ENDING AT 15080,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB30;
CREATE TABLE NEW_ORDERB30
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_030
                INDEX IN ts_newordB_030
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 15081 ENDING AT 15600,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB31;
CREATE TABLE NEW_ORDERB31
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_031
                INDEX IN ts_newordB_031
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 15601 ENDING AT 16120,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB32;
CREATE TABLE NEW_ORDERB32
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_032
                INDEX IN ts_newordB_032
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 16121 ENDING AT 16640,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB33;
CREATE TABLE NEW_ORDERB33
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_033
                INDEX IN ts_newordB_033
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 16641 ENDING AT 17160,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB34;
CREATE TABLE NEW_ORDERB34
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_034
                INDEX IN ts_newordB_034
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 17161 ENDING AT 17680,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB35;
CREATE TABLE NEW_ORDERB35
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_035
                INDEX IN ts_newordB_035
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 17681 ENDING AT 18200,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
```

```
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB36;
CREATE TABLE NEW_ORDERB36
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_036
                INDEX IN ts_newordB_036
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 18201 ENDING AT 18720,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB37;
CREATE TABLE NEW_ORDERB37
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_037
                INDEX IN ts_newordB_037
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 18721 ENDING AT 19240,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB38;
CREATE TABLE NEW_ORDERB38
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_038
                INDEX IN ts_newordB_038
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 19241 ENDING AT 19760,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB39;
CREATE TABLE NEW_ORDERB39
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_039
                INDEX IN ts_newordB_039
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 19761 ENDING AT 20280,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB40;
CREATE TABLE NEW_ORDERB40
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_040
                INDEX IN ts_newordB_040
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 20281 ENDING AT 20800,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB41;
CREATE TABLE NEW_ORDERB41
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_041
                INDEX IN ts_newordB_041
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 20801 ENDING AT 21320,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB42;
CREATE TABLE NEW_ORDERB42
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_042
                INDEX IN ts_newordB_042
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 21321 ENDING AT 21840,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB43;
CREATE TABLE NEW_ORDERB43
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_043
                INDEX IN ts_newordB_043
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 21841 ENDING AT 22360,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB44;
```

```
CREATE TABLE NEW_ORDERB44
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_044
                INDEX IN ts_newordB_044
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 22361 ENDING AT 22880,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB45;
CREATE TABLE NEW_ORDERB45
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_045
                INDEX IN ts_newordB_045
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 22881 ENDING AT 23400,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB46;
CREATE TABLE NEW_ORDERB46
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_046
                INDEX IN ts_newordB_046
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 23401 ENDING AT 23920,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB47;
CREATE TABLE NEW_ORDERB47
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_047
                INDEX IN ts_newordB_047
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 23921 ENDING AT 24440,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB48;
CREATE TABLE NEW_ORDERB48
                (
```

```
                NO_O_ID        INTEGER    NOT NULL,
                NO_D_ID        SMALLINT   NOT NULL,
                NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_048
                INDEX IN ts_newordB_048
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 24441 ENDING AT 24960,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB49;
CREATE TABLE NEW_ORDERB49
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_049
                INDEX IN ts_newordB_049
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 24961 ENDING AT 25480,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB50;
CREATE TABLE NEW_ORDERB50
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_050
                INDEX IN ts_newordB_050
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 25481 ENDING AT 26000,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB51;
CREATE TABLE NEW_ORDERB51
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_051
                INDEX IN ts_newordB_051
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 26001 ENDING AT 26520,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB52;
CREATE TABLE NEW_ORDERB52
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_052
                INDEX IN ts_newordB_052
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 26521 ENDING AT 27040,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB53;
CREATE TABLE NEW_ORDERB53
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_053
                INDEX IN ts_newordB_053
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 27041 ENDING AT 27560,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB54;
CREATE TABLE NEW_ORDERB54
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_054
                INDEX IN ts_newordB_054
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 27561 ENDING AT 28080,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB55;
CREATE TABLE NEW_ORDERB55
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_055
                INDEX IN ts_newordB_055
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 28081 ENDING AT 28600,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB56;
CREATE TABLE NEW_ORDERB56
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_056
                INDEX IN ts_newordB_056
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 28601 ENDING AT 29120,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB57;
CREATE TABLE NEW_ORDERB57
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_057
                INDEX IN ts_newordB_057
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 29121 ENDING AT 29640,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB58;
CREATE TABLE NEW_ORDERB58
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_058
                INDEX IN ts_newordB_058
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 29641 ENDING AT 30160,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB59;
CREATE TABLE NEW_ORDERB59
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_059
                INDEX IN ts_newordB_059
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 30161 ENDING AT 30680,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB60;
CREATE TABLE NEW_ORDERB60
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID        INTEGER    NOT NULL
                )
                IN ts_newordB_060
                INDEX IN ts_newordB_060
```

```
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 30681 ENDING AT 31200,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB61;
CREATE TABLE NEW_ORDERB61
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_061
                INDEX IN ts_newordB_061
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 31201 ENDING AT 31720,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB62;
CREATE TABLE NEW_ORDERB62
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_062
                INDEX IN ts_newordB_062
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 31721 ENDING AT 32240,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB63;
CREATE TABLE NEW_ORDERB63
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_063
                INDEX IN ts_newordB_063
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 32241 ENDING AT 32760,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB64;
CREATE TABLE NEW_ORDERB64
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_064
                INDEX IN ts_newordB_064
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 32761 ENDING AT 33280,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB65;
CREATE TABLE NEW_ORDERB65
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_065
                INDEX IN ts_newordB_065
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 33281 ENDING AT 33800,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB66;
CREATE TABLE NEW_ORDERB66
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_066
                INDEX IN ts_newordB_066
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 33801 ENDING AT 34320,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB67;
CREATE TABLE NEW_ORDERB67
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_067
                INDEX IN ts_newordB_067
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 34321 ENDING AT 34840,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB68;
CREATE TABLE NEW_ORDERB68
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_068
                INDEX IN ts_newordB_068
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 34841 ENDING AT 35360,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB69;
CREATE TABLE NEW_ORDERB69
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_069
                INDEX IN ts_newordB_069
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 35361 ENDING AT 35880,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB70;
CREATE TABLE NEW_ORDERB70
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_070
                INDEX IN ts_newordB_070
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 35881 ENDING AT 36400,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB71;
CREATE TABLE NEW_ORDERB71
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_071
                INDEX IN ts_newordB_071
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 36401 ENDING AT 36920,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB72;
CREATE TABLE NEW_ORDERB72
                (
                 NO_O_ID        INTEGER    NOT NULL,
                 NO_D_ID        SMALLINT   NOT NULL,
                 NO_W_ID         INTEGER    NOT NULL
                )
                IN ts_newordB_072
                INDEX IN ts_newordB_072
                ORGANIZE BY KEY SEQUENCE (
                 NO_W_ID STARTING FROM 36921 ENDING AT 37440,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB73;
CREATE TABLE NEW_ORDERB73
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_073
        INDEX IN ts_newordB_073
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 37441 ENDING AT 37960,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB74;
CREATE TABLE NEW_ORDERB74
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_074
        INDEX IN ts_newordB_074
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 37961 ENDING AT 38480,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB75;
CREATE TABLE NEW_ORDERB75
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_075
        INDEX IN ts_newordB_075
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 38481 ENDING AT 39000,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB76;
CREATE TABLE NEW_ORDERB76
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_076
        INDEX IN ts_newordB_076
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 39001 ENDING AT 39520,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB77;
CREATE TABLE NEW_ORDERB77
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_077
        INDEX IN ts_newordB_077
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 39521 ENDING AT 40040,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB78;
CREATE TABLE NEW_ORDERB78
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_078
        INDEX IN ts_newordB_078
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 40041 ENDING AT 40560,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB79;
CREATE TABLE NEW_ORDERB79
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_079
        INDEX IN ts_newordB_079
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 40561 ENDING AT 41080,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB80;
CREATE TABLE NEW_ORDERB80
        (
         NO_O_ID        INTEGER    NOT NULL,
         NO_D_ID        SMALLINT   NOT NULL,
         NO_W_ID        INTEGER    NOT NULL
        )
        IN ts_newordB_080
        INDEX IN ts_newordB_080
        ORGANIZE BY KEY SEQUENCE (
         NO_W_ID STARTING FROM 41081 ENDING AT 41600,
         NO_D_ID STARTING FROM 1 ENDING AT 10,
         NO_O_ID STARTING FROM 3676 ENDING AT 5451
        )
        ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_ORDERS.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDERS1;
CREATE TABLE ORDERS1
        (
         O_C_ID         INTEGER     NOT NULL,
         O_ENTRY_D      TIMESTAMP   NOT NULL,
         O_CARRIER_ID   SMALLINT    NOT NULL,
         O_OL_CNT       SMALLINT    NOT NULL,
         O_ALL_LOCAL    SMALLINT    NOT NULL,
         O_ID           INTEGER     NOT NULL,
         O_W_ID         INTEGER     NOT NULL,
         O_D_ID         SMALLINT    NOT NULL
        )
        IN ts_order_001
        INDEX IN is_order_001
        ORGANIZE BY KEY SEQUENCE (
         O_ID STARTING FROM 1 ENDING AT 3675,
         O_W_ID STARTING FROM 1 ENDING AT 520,
         O_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS2;
CREATE TABLE ORDERS2
        (
         O_C_ID         INTEGER     NOT NULL,
         O_ENTRY_D      TIMESTAMP   NOT NULL,
         O_CARRIER_ID   SMALLINT    NOT NULL,
         O_OL_CNT       SMALLINT    NOT NULL,
         O_ALL_LOCAL    SMALLINT    NOT NULL,
         O_ID           INTEGER     NOT NULL,
         O_W_ID         INTEGER     NOT NULL,
         O_D_ID         SMALLINT    NOT NULL
        )
        IN ts_order_002
        INDEX IN is_order_002
        ORGANIZE BY KEY SEQUENCE (
         O_ID STARTING FROM 1 ENDING AT 3675,
         O_W_ID STARTING FROM 521 ENDING AT 1040,
         O_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS3;
CREATE TABLE ORDERS3
        (
         O_C_ID         INTEGER     NOT NULL,
         O_ENTRY_D      TIMESTAMP   NOT NULL,
         O_CARRIER_ID   SMALLINT    NOT NULL,
         O_OL_CNT       SMALLINT    NOT NULL,
         O_ALL_LOCAL    SMALLINT    NOT NULL,
         O_ID           INTEGER     NOT NULL,
         O_W_ID         INTEGER     NOT NULL,
         O_D_ID         SMALLINT    NOT NULL
        )
        IN ts_order_003
        INDEX IN is_order_003
        ORGANIZE BY KEY SEQUENCE (
         O_ID STARTING FROM 1 ENDING AT 3675,
         O_W_ID STARTING FROM 1041 ENDING AT 1560,
         O_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS4;
CREATE TABLE ORDERS4
        (
```

```
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_004
               INDEX IN is_order_004
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 1561 ENDING AT 2080,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS5;
CREATE TABLE ORDERS5
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_005
               INDEX IN is_order_005
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 2081 ENDING AT 2600,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS6;
CREATE TABLE ORDERS6
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_006
               INDEX IN is_order_006
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 2601 ENDING AT 3120,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS7;
CREATE TABLE ORDERS7
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_007
               INDEX IN is_order_007
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 3121 ENDING AT 3640,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS8;
CREATE TABLE ORDERS8
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_008
               INDEX IN is_order_008
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 3641 ENDING AT 4160,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS9;
CREATE TABLE ORDERS9
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_009
               INDEX IN is_order_009
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 4161 ENDING AT 4680,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS10;
CREATE TABLE ORDERS10
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_010
               INDEX IN is_order_010
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 4681 ENDING AT 5200,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS11;
CREATE TABLE ORDERS11
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_011
               INDEX IN is_order_011
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 5201 ENDING AT 5720,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS12;
CREATE TABLE ORDERS12
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_012
               INDEX IN is_order_012
               ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 5721 ENDING AT 6240,
                O_D_ID STARTING FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS13;
CREATE TABLE ORDERS13
               (
                O_C_ID        INTEGER     NOT NULL,
                O_ENTRY_D     TIMESTAMP   NOT NULL,
                O_CARRIER_ID  SMALLINT    NOT NULL,
                O_OL_CNT      SMALLINT    NOT NULL,
                O_ALL_LOCAL   SMALLINT    NOT NULL,
                O_ID          INTEGER     NOT NULL,
                O_W_ID        INTEGER     NOT NULL,
                O_D_ID        SMALLINT    NOT NULL
               )
               IN ts_order_013
               INDEX IN is_order_013
               ORGANIZE BY KEY SEQUENCE (
```

```
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 6241 ENDING AT 6760,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS14;
CREATE TABLE ORDERS14
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_014
                INDEX IN is_order_014
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 6761 ENDING AT 7280,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS15;
CREATE TABLE ORDERS15
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_015
                INDEX IN is_order_015
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 7281 ENDING AT 7800,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS16;
CREATE TABLE ORDERS16
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_016
                INDEX IN is_order_016
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 7801 ENDING AT 8320,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS17;
CREATE TABLE ORDERS17
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_017
                INDEX IN is_order_017
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 8321 ENDING AT 8840,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS18;
CREATE TABLE ORDERS18
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_018
                INDEX IN is_order_018
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 8841 ENDING AT 9360,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS19;
CREATE TABLE ORDERS19
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_019
                INDEX IN is_order_019
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 9361 ENDING AT 9880,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS20;
CREATE TABLE ORDERS20
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_020
                INDEX IN is_order_020
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 9881 ENDING AT 10400,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS21;
CREATE TABLE ORDERS21
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_021
                INDEX IN is_order_021
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 10401 ENDING AT 10920,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS22;
CREATE TABLE ORDERS22
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
                O_CARRIER_ID    SMALLINT    NOT NULL,
                O_OL_CNT        SMALLINT    NOT NULL,
                O_ALL_LOCAL     SMALLINT    NOT NULL,
                O_ID            INTEGER     NOT NULL,
                O_W_ID          INTEGER     NOT NULL,
                O_D_ID          SMALLINT    NOT NULL
                )
                IN ts_order_022
                INDEX IN is_order_022
                ORGANIZE BY KEY SEQUENCE (
                O_ID STARTING FROM 1 ENDING AT 3675,
                O_W_ID STARTING FROM 10921 ENDING AT 11440,
                O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS23;
CREATE TABLE ORDERS23
                (
                O_C_ID          INTEGER     NOT NULL,
                O_ENTRY_D       TIMESTAMP   NOT NULL,
```

```
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_023
INDEX IN is_order_023
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 11441 ENDING AT 11960,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS24;
CREATE TABLE ORDERS24
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_024
INDEX IN is_order_024
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 11961 ENDING AT 12480,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS25;
CREATE TABLE ORDERS25
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_025
INDEX IN is_order_025
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 12481 ENDING AT 13000,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS26;
CREATE TABLE ORDERS26
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_026
INDEX IN is_order_026
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 13001 ENDING AT 13520,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS27;
CREATE TABLE ORDERS27
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_027
INDEX IN is_order_027
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 13521 ENDING AT 14040,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS28;
CREATE TABLE ORDERS28
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_028
INDEX IN is_order_028
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 14041 ENDING AT 14560,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS29;
CREATE TABLE ORDERS29
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_029
INDEX IN is_order_029
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 14561 ENDING AT 15080,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS30;
CREATE TABLE ORDERS30
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_030
INDEX IN is_order_030
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 15081 ENDING AT 15600,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS31;
CREATE TABLE ORDERS31
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_031
INDEX IN is_order_031
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 15601 ENDING AT 16120,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS32;
CREATE TABLE ORDERS32
(
 O_C_ID         INTEGER     NOT NULL,
 O_ENTRY_D      TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_032
INDEX IN is_order_032
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 16121 ENDING AT 16640,
```

```
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS33;
CREATE TABLE ORDERS33
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_033
INDEX IN is_order_033
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 16641 ENDING AT 17160,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS34;
CREATE TABLE ORDERS34
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_034
INDEX IN is_order_034
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 17161 ENDING AT 17680,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS35;
CREATE TABLE ORDERS35
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_035
INDEX IN is_order_035
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 17681 ENDING AT 18200,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS36;
CREATE TABLE ORDERS36
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_036
INDEX IN is_order_036
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 18201 ENDING AT 18720,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS37;
CREATE TABLE ORDERS37
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_037
INDEX IN is_order_037
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 18721 ENDING AT 19240,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS38;
CREATE TABLE ORDERS38
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_038
INDEX IN is_order_038
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 19241 ENDING AT 19760,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS39;
CREATE TABLE ORDERS39
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_039
INDEX IN is_order_039
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 19761 ENDING AT 20280,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS40;
CREATE TABLE ORDERS40
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_040
INDEX IN is_order_040
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 20281 ENDING AT 20800,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS41;
CREATE TABLE ORDERS41
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_041
INDEX IN is_order_041
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 20801 ENDING AT 21320,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS42;
CREATE TABLE ORDERS42
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
```

```
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_042
INDEX IN is_order_042
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 21321 ENDING AT 21840,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS43;
CREATE TABLE ORDERS43
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_043
INDEX IN is_order_043
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 21841 ENDING AT 22360,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS44;
CREATE TABLE ORDERS44
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_044
INDEX IN is_order_044
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 22361 ENDING AT 22880,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS45;
CREATE TABLE ORDERS45
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_045
INDEX IN is_order_045
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 22881 ENDING AT 23400,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS46;
CREATE TABLE ORDERS46
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_046
INDEX IN is_order_046
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 23401 ENDING AT 23920,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS47;
CREATE TABLE ORDERS47
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_047
INDEX IN is_order_047
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 23921 ENDING AT 24440,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS48;
CREATE TABLE ORDERS48
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_048
INDEX IN is_order_048
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 24441 ENDING AT 24960,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS49;
CREATE TABLE ORDERS49
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_049
INDEX IN is_order_049
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 24961 ENDING AT 25480,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS50;
CREATE TABLE ORDERS50
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_050
INDEX IN is_order_050
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 25481 ENDING AT 26000,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS51;
CREATE TABLE ORDERS51
(
 O_C_ID          INTEGER     NOT NULL,
 O_ENTRY_D       TIMESTAMP   NOT NULL,
 O_CARRIER_ID    SMALLINT    NOT NULL,
 O_OL_CNT        SMALLINT    NOT NULL,
 O_ALL_LOCAL     SMALLINT    NOT NULL,
 O_ID            INTEGER     NOT NULL,
 O_W_ID          INTEGER     NOT NULL,
 O_D_ID          SMALLINT    NOT NULL
)
IN ts_order_051
INDEX IN is_order_051
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 26001 ENDING AT 26520,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
```

```
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS52;
CREATE TABLE ORDERS52
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_052
                INDEX IN is_order_052
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 26521 ENDING AT 27040,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS53;
CREATE TABLE ORDERS53
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_053
                INDEX IN is_order_053
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 27041 ENDING AT 27560,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS54;
CREATE TABLE ORDERS54
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_054
                INDEX IN is_order_054
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 27561 ENDING AT 28080,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS55;
CREATE TABLE ORDERS55
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_055
                INDEX IN is_order_055
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 28081 ENDING AT 28600,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS56;
CREATE TABLE ORDERS56
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_056
                INDEX IN is_order_056
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 28601 ENDING AT 29120,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS57;
CREATE TABLE ORDERS57
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_057
                INDEX IN is_order_057
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 29121 ENDING AT 29640,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS58;
CREATE TABLE ORDERS58
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_058
                INDEX IN is_order_058
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 29641 ENDING AT 30160,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS59;
CREATE TABLE ORDERS59
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_059
                INDEX IN is_order_059
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 30161 ENDING AT 30680,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS60;
CREATE TABLE ORDERS60
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_060
                INDEX IN is_order_060
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 30681 ENDING AT 31200,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS61;
CREATE TABLE ORDERS61
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
```

```
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_061
INDEX IN is_order_061
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 31201 ENDING AT 31720,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS62;
CREATE TABLE ORDERS62
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_062
INDEX IN is_order_062
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 31721 ENDING AT 32240,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS63;
CREATE TABLE ORDERS63
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_063
INDEX IN is_order_063
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 32241 ENDING AT 32760,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS64;
CREATE TABLE ORDERS64
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_064
INDEX IN is_order_064
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 32761 ENDING AT 33280,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS65;
CREATE TABLE ORDERS65
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_065
INDEX IN is_order_065
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 33281 ENDING AT 33800,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS66;
CREATE TABLE ORDERS66
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_066
INDEX IN is_order_066
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 33801 ENDING AT 34320,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS67;
CREATE TABLE ORDERS67
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_067
INDEX IN is_order_067
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 34321 ENDING AT 34840,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS68;
CREATE TABLE ORDERS68
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_068
INDEX IN is_order_068
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 34841 ENDING AT 35360,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS69;
CREATE TABLE ORDERS69
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_069
INDEX IN is_order_069
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 35361 ENDING AT 35880,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS70;
CREATE TABLE ORDERS70
(
 O_C_ID        INTEGER     NOT NULL,
 O_ENTRY_D     TIMESTAMP   NOT NULL,
 O_CARRIER_ID  SMALLINT    NOT NULL,
 O_OL_CNT      SMALLINT    NOT NULL,
 O_ALL_LOCAL   SMALLINT    NOT NULL,
 O_ID          INTEGER     NOT NULL,
 O_W_ID        INTEGER     NOT NULL,
 O_D_ID        SMALLINT    NOT NULL
)
IN ts_order_070
INDEX IN is_order_070
ORGANIZE BY KEY SEQUENCE (
 O_ID STARTING FROM 1 ENDING AT 3675,
 O_W_ID STARTING FROM 35881 ENDING AT 36400,
 O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE ORDERS71;
CREATE TABLE ORDERS71
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_071
                INDEX IN is_order_071
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 36401 ENDING AT 36920,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS72;
CREATE TABLE ORDERS72
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_072
                INDEX IN is_order_072
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 36921 ENDING AT 37440,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS73;
CREATE TABLE ORDERS73
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_073
                INDEX IN is_order_073
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 37441 ENDING AT 37960,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS74;
CREATE TABLE ORDERS74
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_074
                INDEX IN is_order_074
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 37961 ENDING AT 38480,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS75;
CREATE TABLE ORDERS75
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_075
                INDEX IN is_order_075
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 38481 ENDING AT 39000,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS76;
CREATE TABLE ORDERS76
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_076
                INDEX IN is_order_076
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 39001 ENDING AT 39520,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS77;
CREATE TABLE ORDERS77
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_077
                INDEX IN is_order_077
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 39521 ENDING AT 40040,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS78;
CREATE TABLE ORDERS78
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_078
                INDEX IN is_order_078
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 40041 ENDING AT 40560,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS79;
CREATE TABLE ORDERS79
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
                )
                IN ts_order_079
                INDEX IN is_order_079
                ORGANIZE BY KEY SEQUENCE (
                 O_ID STARTING FROM 1 ENDING AT 3675,
                 O_W_ID STARTING FROM 40561 ENDING AT 41080,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS80;
CREATE TABLE ORDERS80
                (
                 O_C_ID         INTEGER     NOT NULL,
                 O_ENTRY_D      TIMESTAMP   NOT NULL,
                 O_CARRIER_ID   SMALLINT    NOT NULL,
                 O_OL_CNT       SMALLINT    NOT NULL,
                 O_ALL_LOCAL    SMALLINT    NOT NULL,
                 O_ID           INTEGER     NOT NULL,
                 O_W_ID         INTEGER     NOT NULL,
                 O_D_ID         SMALLINT    NOT NULL
```

```
            )
            IN ts_order_080
            INDEX IN is_order_080
            ORGANIZE BY KEY SEQUENCE (
             O_ID STARTING FROM 1 ENDING AT 3675,
             O_W_ID STARTING FROM 41081 ENDING AT 41600,
             O_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_ORDER_LINE.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE1;
CREATE TABLE ORDER_LINE1
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_001
            INDEX IN ts_orderline_001
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 1 ENDING AT 520,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE2;
CREATE TABLE ORDER_LINE2
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_002
            INDEX IN ts_orderline_002
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 521 ENDING AT 1040,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE3;
CREATE TABLE ORDER_LINE3
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_003
            INDEX IN ts_orderline_003
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 1041 ENDING AT 1560,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE4;
CREATE TABLE ORDER_LINE4
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_004
            INDEX IN ts_orderline_004
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 1561 ENDING AT 2080,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE5;
CREATE TABLE ORDER_LINE5
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_005
            INDEX IN ts_orderline_005
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 2081 ENDING AT 2600,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE6;
CREATE TABLE ORDER_LINE6
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_006
            INDEX IN ts_orderline_006
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 2601 ENDING AT 3120,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE7;
CREATE TABLE ORDER_LINE7
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_007
            INDEX IN ts_orderline_007
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 3121 ENDING AT 3640,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE8;
CREATE TABLE ORDER_LINE8
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_008
            INDEX IN ts_orderline_008
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 3641 ENDING AT 4160,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
```

```
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE9;
CREATE TABLE ORDER_LINE9
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_009
        INDEX IN ts_orderline_009
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 4161 ENDING AT 4680,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE10;
CREATE TABLE ORDER_LINE10
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_010
        INDEX IN ts_orderline_010
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 4681 ENDING AT 5200,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE11;
CREATE TABLE ORDER_LINE11
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_011
        INDEX IN ts_orderline_011
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 5201 ENDING AT 5720,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE12;
CREATE TABLE ORDER_LINE12
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_012
        INDEX IN ts_orderline_012
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 5721 ENDING AT 6240,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE13;
CREATE TABLE ORDER_LINE13
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_013
        INDEX IN ts_orderline_013
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 6241 ENDING AT 6760,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE14;
CREATE TABLE ORDER_LINE14
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_014
        INDEX IN ts_orderline_014
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 6761 ENDING AT 7280,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE15;
CREATE TABLE ORDER_LINE15
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_015
        INDEX IN ts_orderline_015
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 7281 ENDING AT 7800,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE16;
CREATE TABLE ORDER_LINE16
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_016
        INDEX IN ts_orderline_016
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 7801 ENDING AT 8320,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE17;
CREATE TABLE ORDER_LINE17
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
```

```
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_017
               INDEX IN ts_orderline_017
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 8321 ENDING AT 8840,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE18;
CREATE TABLE ORDER_LINE18
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_018
               INDEX IN ts_orderline_018
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 8841 ENDING AT 9360,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE19;
CREATE TABLE ORDER_LINE19
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_019
               INDEX IN ts_orderline_019
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 9361 ENDING AT 9880,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE20;
CREATE TABLE ORDER_LINE20
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_020
               INDEX IN ts_orderline_020
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 9881 ENDING AT 10400,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE21;
CREATE TABLE ORDER_LINE21
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_021
               INDEX IN ts_orderline_021
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 10401 ENDING AT 10920,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE22;
CREATE TABLE ORDER_LINE22
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_022
               INDEX IN ts_orderline_022
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 10921 ENDING AT 11440,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE23;
CREATE TABLE ORDER_LINE23
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_023
               INDEX IN ts_orderline_023
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 11441 ENDING AT 11960,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE24;
CREATE TABLE ORDER_LINE24
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_024
               INDEX IN ts_orderline_024
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 11961 ENDING AT 12480,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE25;
CREATE TABLE ORDER_LINE25
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_025
               INDEX IN ts_orderline_025
```

```
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 12481 ENDING AT 13000,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE26;
CREATE TABLE ORDER_LINE26
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_026
            INDEX IN ts_orderline_026
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 13001 ENDING AT 13520,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE27;
CREATE TABLE ORDER_LINE27
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_027
            INDEX IN ts_orderline_027
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 13521 ENDING AT 14040,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE28;
CREATE TABLE ORDER_LINE28
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_028
            INDEX IN ts_orderline_028
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 14041 ENDING AT 14560,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE29;
CREATE TABLE ORDER_LINE29
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_029
            INDEX IN ts_orderline_029
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 14561 ENDING AT 15080,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE30;
CREATE TABLE ORDER_LINE30
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_030
            INDEX IN ts_orderline_030
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 15081 ENDING AT 15600,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE31;
CREATE TABLE ORDER_LINE31
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_031
            INDEX IN ts_orderline_031
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 15601 ENDING AT 16120,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE32;
CREATE TABLE ORDER_LINE32
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_032
            INDEX IN ts_orderline_032
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 16121 ENDING AT 16640,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE33;
CREATE TABLE ORDER_LINE33
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_033
            INDEX IN ts_orderline_033
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 16641 ENDING AT 17160,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE34;
```

```
CREATE TABLE ORDER_LINE34
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_034
        INDEX IN ts_orderline_034
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 17161 ENDING AT 17680,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE35;
CREATE TABLE ORDER_LINE35
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_035
        INDEX IN ts_orderline_035
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 17681 ENDING AT 18200,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE36;
CREATE TABLE ORDER_LINE36
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_036
        INDEX IN ts_orderline_036
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 18201 ENDING AT 18720,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE37;
CREATE TABLE ORDER_LINE37
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_037
        INDEX IN ts_orderline_037
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 18721 ENDING AT 19240,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE38;
CREATE TABLE ORDER_LINE38
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_038
        INDEX IN ts_orderline_038
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 19241 ENDING AT 19760,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE39;
CREATE TABLE ORDER_LINE39
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_039
        INDEX IN ts_orderline_039
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 19761 ENDING AT 20280,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE40;
CREATE TABLE ORDER_LINE40
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_040
        INDEX IN ts_orderline_040
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 20281 ENDING AT 20800,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE41;
CREATE TABLE ORDER_LINE41
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_041
        INDEX IN ts_orderline_041
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 20801 ENDING AT 21320,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE42;
CREATE TABLE ORDER_LINE42
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
```

```
    OL_W_ID         INTEGER      NOT NULL,
    OL_NUMBER       SMALLINT     NOT NULL
   )
   IN ts_orderline_042
   INDEX IN ts_orderline_042
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 21321 ENDING AT 21840,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE43;
CREATE TABLE ORDER_LINE43
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_043
   INDEX IN ts_orderline_043
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 21841 ENDING AT 22360,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE44;
CREATE TABLE ORDER_LINE44
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_044
   INDEX IN ts_orderline_044
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 22361 ENDING AT 22880,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE45;
CREATE TABLE ORDER_LINE45
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_045
   INDEX IN ts_orderline_045
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 22881 ENDING AT 23400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE46;
CREATE TABLE ORDER_LINE46
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_046
   INDEX IN ts_orderline_046
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 23401 ENDING AT 23920,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE47;
CREATE TABLE ORDER_LINE47
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_047
   INDEX IN ts_orderline_047
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 23921 ENDING AT 24440,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE48;
CREATE TABLE ORDER_LINE48
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_048
   INDEX IN ts_orderline_048
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 24441 ENDING AT 24960,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE49;
CREATE TABLE ORDER_LINE49
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_049
   INDEX IN ts_orderline_049
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 24961 ENDING AT 25480,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
   )
   ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE50;
CREATE TABLE ORDER_LINE50
   (
    OL_DELIVERY_D    TIMESTAMP    NOT NULL,
    OL_AMOUNT        DECIMAL(6,2) NOT NULL,
    OL_I_ID          INTEGER      NOT NULL,
    OL_SUPPLY_W_ID   INTEGER      NOT NULL,
    OL_QUANTITY      SMALLINT     NOT NULL,
    OL_DIST_INFO     CHAR(24)     NOT NULL,
    OL_O_ID          INTEGER      NOT NULL,
    OL_D_ID          SMALLINT     NOT NULL,
    OL_W_ID          INTEGER      NOT NULL,
    OL_NUMBER        SMALLINT     NOT NULL
   )
   IN ts_orderline_050
   INDEX IN ts_orderline_050
   ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 25481 ENDING AT 26000,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
```

```
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE51;
CREATE TABLE ORDER_LINE51
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_051
        INDEX IN ts_orderline_051
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 26001 ENDING AT 26520,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE52;
CREATE TABLE ORDER_LINE52
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_052
        INDEX IN ts_orderline_052
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 26521 ENDING AT 27040,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE53;
CREATE TABLE ORDER_LINE53
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_053
        INDEX IN ts_orderline_053
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 27041 ENDING AT 27560,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE54;
CREATE TABLE ORDER_LINE54
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_054
        INDEX IN ts_orderline_054
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 27561 ENDING AT 28080,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE55;
CREATE TABLE ORDER_LINE55
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_055
        INDEX IN ts_orderline_055
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 28081 ENDING AT 28600,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE56;
CREATE TABLE ORDER_LINE56
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_056
        INDEX IN ts_orderline_056
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 28601 ENDING AT 29120,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE57;
CREATE TABLE ORDER_LINE57
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_057
        INDEX IN ts_orderline_057
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 29121 ENDING AT 29640,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE58;
CREATE TABLE ORDER_LINE58
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_058
        INDEX IN ts_orderline_058
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 29641 ENDING AT 30160,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE59;
CREATE TABLE ORDER_LINE59
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
```

```
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_059
               INDEX IN ts_orderline_059
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 30161 ENDING AT 30680,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE60;
CREATE TABLE ORDER_LINE60
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_060
               INDEX IN ts_orderline_060
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 30681 ENDING AT 31200,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE61;
CREATE TABLE ORDER_LINE61
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_061
               INDEX IN ts_orderline_061
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 31201 ENDING AT 31720,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE62;
CREATE TABLE ORDER_LINE62
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_062
               INDEX IN ts_orderline_062
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 31721 ENDING AT 32240,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE63;
CREATE TABLE ORDER_LINE63
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_063
               INDEX IN ts_orderline_063
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 32241 ENDING AT 32760,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE64;
CREATE TABLE ORDER_LINE64
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_064
               INDEX IN ts_orderline_064
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 32761 ENDING AT 33280,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE65;
CREATE TABLE ORDER_LINE65
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_065
               INDEX IN ts_orderline_065
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 33281 ENDING AT 33800,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE66;
CREATE TABLE ORDER_LINE66
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_066
               INDEX IN ts_orderline_066
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 33801 ENDING AT 34320,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3675,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE67;
CREATE TABLE ORDER_LINE67
               (
                OL_DELIVERY_D    TIMESTAMP    NOT NULL,
                OL_AMOUNT        DECIMAL(6,2) NOT NULL,
                OL_I_ID          INTEGER      NOT NULL,
                OL_SUPPLY_W_ID   INTEGER      NOT NULL,
                OL_QUANTITY      SMALLINT     NOT NULL,
                OL_DIST_INFO     CHAR(24)     NOT NULL,
                OL_O_ID          INTEGER      NOT NULL,
                OL_D_ID          SMALLINT     NOT NULL,
                OL_W_ID          INTEGER      NOT NULL,
                OL_NUMBER        SMALLINT     NOT NULL
               )
               IN ts_orderline_067
               INDEX IN ts_orderline_067
```

```
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 34321 ENDING AT 34840,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE68;
CREATE TABLE ORDER_LINE68
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_068
            INDEX IN ts_orderline_068
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 34841 ENDING AT 35360,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE69;
CREATE TABLE ORDER_LINE69
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_069
            INDEX IN ts_orderline_069
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 35361 ENDING AT 35880,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE70;
CREATE TABLE ORDER_LINE70
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_070
            INDEX IN ts_orderline_070
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 35881 ENDING AT 36400,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE71;
CREATE TABLE ORDER_LINE71
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_071
            INDEX IN ts_orderline_071
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 36401 ENDING AT 36920,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE72;
CREATE TABLE ORDER_LINE72
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_072
            INDEX IN ts_orderline_072
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 36921 ENDING AT 37440,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE73;
CREATE TABLE ORDER_LINE73
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_073
            INDEX IN ts_orderline_073
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 37441 ENDING AT 37960,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE74;
CREATE TABLE ORDER_LINE74
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_074
            INDEX IN ts_orderline_074
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 37961 ENDING AT 38480,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE75;
CREATE TABLE ORDER_LINE75
            (
             OL_DELIVERY_D    TIMESTAMP    NOT NULL,
             OL_AMOUNT        DECIMAL(6,2) NOT NULL,
             OL_I_ID          INTEGER      NOT NULL,
             OL_SUPPLY_W_ID   INTEGER      NOT NULL,
             OL_QUANTITY      SMALLINT     NOT NULL,
             OL_DIST_INFO     CHAR(24)     NOT NULL,
             OL_O_ID          INTEGER      NOT NULL,
             OL_D_ID          SMALLINT     NOT NULL,
             OL_W_ID          INTEGER      NOT NULL,
             OL_NUMBER        SMALLINT     NOT NULL
            )
            IN ts_orderline_075
            INDEX IN ts_orderline_075
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 38481 ENDING AT 39000,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3675,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE76;
```

```
CREATE TABLE ORDER_LINE76
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_076
        INDEX IN ts_orderline_076
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 39001 ENDING AT 39520,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE77;
CREATE TABLE ORDER_LINE77
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_077
        INDEX IN ts_orderline_077
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 39521 ENDING AT 40040,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE78;
CREATE TABLE ORDER_LINE78
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_078
        INDEX IN ts_orderline_078
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 40041 ENDING AT 40560,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE79;
CREATE TABLE ORDER_LINE79
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_079
        INDEX IN ts_orderline_079
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 40561 ENDING AT 41080,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE80;
CREATE TABLE ORDER_LINE80
        (
         OL_DELIVERY_D    TIMESTAMP    NOT NULL,
         OL_AMOUNT        DECIMAL(6,2) NOT NULL,
         OL_I_ID          INTEGER      NOT NULL,
         OL_SUPPLY_W_ID   INTEGER      NOT NULL,
         OL_QUANTITY      SMALLINT     NOT NULL,
         OL_DIST_INFO     CHAR(24)     NOT NULL,
         OL_O_ID          INTEGER      NOT NULL,
         OL_D_ID          SMALLINT     NOT NULL,
         OL_W_ID          INTEGER      NOT NULL,
         OL_NUMBER        SMALLINT     NOT NULL
        )
        IN ts_orderline_080
        INDEX IN ts_orderline_080
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 41081 ENDING AT 41600,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3675,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_STOCK.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK1;
CREATE TABLE STOCK1
        (
         S_REMOTE_CNT    INTEGER     NOT NULL,
         S_QUANTITY      INTEGER     NOT NULL,
         S_ORDER_CNT     INTEGER     NOT NULL,
         S_YTD           INTEGER     NOT NULL,
         S_DATA          VARCHAR(50) NOT NULL,
         S_DIST_01       CHAR(24)    NOT NULL,
         S_DIST_02       CHAR(24)    NOT NULL,
         S_DIST_03       CHAR(24)    NOT NULL,
         S_DIST_04       CHAR(24)    NOT NULL,
         S_DIST_05       CHAR(24)    NOT NULL,
         S_DIST_06       CHAR(24)    NOT NULL,
         S_DIST_07       CHAR(24)    NOT NULL,
         S_DIST_08       CHAR(24)    NOT NULL,
         S_DIST_09       CHAR(24)    NOT NULL,
         S_DIST_10       CHAR(24)    NOT NULL,
         S_I_ID          INTEGER     NOT NULL,
         S_W_ID          INTEGER     NOT NULL
        )
        IN ts_stock_001
        INDEX IN ts_stock_001
        ORGANIZE BY KEY SEQUENCE (
         S_I_ID STARTING FROM 1 ENDING AT 100000,
         S_W_ID STARTING FROM 1 ENDING AT 520
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK2;
CREATE TABLE STOCK2
        (
         S_REMOTE_CNT    INTEGER     NOT NULL,
         S_QUANTITY      INTEGER     NOT NULL,
         S_ORDER_CNT     INTEGER     NOT NULL,
         S_YTD           INTEGER     NOT NULL,
         S_DATA          VARCHAR(50) NOT NULL,
         S_DIST_01       CHAR(24)    NOT NULL,
         S_DIST_02       CHAR(24)    NOT NULL,
         S_DIST_03       CHAR(24)    NOT NULL,
         S_DIST_04       CHAR(24)    NOT NULL,
         S_DIST_05       CHAR(24)    NOT NULL,
         S_DIST_06       CHAR(24)    NOT NULL,
         S_DIST_07       CHAR(24)    NOT NULL,
         S_DIST_08       CHAR(24)    NOT NULL,
         S_DIST_09       CHAR(24)    NOT NULL,
         S_DIST_10       CHAR(24)    NOT NULL,
         S_I_ID          INTEGER     NOT NULL,
         S_W_ID          INTEGER     NOT NULL
        )
        IN ts_stock_002
        INDEX IN ts_stock_002
        ORGANIZE BY KEY SEQUENCE (
         S_I_ID STARTING FROM 1 ENDING AT 100000,
         S_W_ID STARTING FROM 521 ENDING AT 1040
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK3;
CREATE TABLE STOCK3
        (
         S_REMOTE_CNT    INTEGER     NOT NULL,
         S_QUANTITY      INTEGER     NOT NULL,
         S_ORDER_CNT     INTEGER     NOT NULL,
         S_YTD           INTEGER     NOT NULL,
         S_DATA          VARCHAR(50) NOT NULL,
         S_DIST_01       CHAR(24)    NOT NULL,
         S_DIST_02       CHAR(24)    NOT NULL,
         S_DIST_03       CHAR(24)    NOT NULL,
         S_DIST_04       CHAR(24)    NOT NULL,
         S_DIST_05       CHAR(24)    NOT NULL,
         S_DIST_06       CHAR(24)    NOT NULL,
         S_DIST_07       CHAR(24)    NOT NULL,
         S_DIST_08       CHAR(24)    NOT NULL,
         S_DIST_09       CHAR(24)    NOT NULL,
         S_DIST_10       CHAR(24)    NOT NULL,
         S_I_ID          INTEGER     NOT NULL,
         S_W_ID          INTEGER     NOT NULL
        )
        IN ts_stock_003
        INDEX IN ts_stock_003
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 1041 ENDING AT 1560
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK4;
CREATE TABLE STOCK4
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_004
            INDEX IN ts_stock_004
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 1561 ENDING AT 2080
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK5;
CREATE TABLE STOCK5
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_005
            INDEX IN ts_stock_005
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 2081 ENDING AT 2600
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK6;
CREATE TABLE STOCK6
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_006
            INDEX IN ts_stock_006
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 2601 ENDING AT 3120
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK7;
CREATE TABLE STOCK7
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_007
            INDEX IN ts_stock_007
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 3121 ENDING AT 3640
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK8;
CREATE TABLE STOCK8
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_008
            INDEX IN ts_stock_008
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 3641 ENDING AT 4160
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK9;
CREATE TABLE STOCK9
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_009
            INDEX IN ts_stock_009
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 4161 ENDING AT 4680
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK10;
CREATE TABLE STOCK10
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_010
            INDEX IN ts_stock_010
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 4681 ENDING AT 5200
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK11;
CREATE TABLE STOCK11
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_011
            INDEX IN ts_stock_011
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 5201 ENDING AT 5720
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK12;
CREATE TABLE STOCK12
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_012
            INDEX IN ts_stock_012
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 5721 ENDING AT 6240
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK13;
CREATE TABLE STOCK13
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_013
            INDEX IN ts_stock_013
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 6241 ENDING AT 6760
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK14;
CREATE TABLE STOCK14
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_014
            INDEX IN ts_stock_014
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 6761 ENDING AT 7280
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK15;
CREATE TABLE STOCK15
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_015
            INDEX IN ts_stock_015
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 7281 ENDING AT 7800
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK16;
CREATE TABLE STOCK16
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_016
            INDEX IN ts_stock_016
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 7801 ENDING AT 8320
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK17;
CREATE TABLE STOCK17
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_017
            INDEX IN ts_stock_017
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 8321 ENDING AT 8840
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK18;
CREATE TABLE STOCK18
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_018
            INDEX IN ts_stock_018
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 8841 ENDING AT 9360
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK19;
CREATE TABLE STOCK19
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_019
            INDEX IN ts_stock_019
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 9361 ENDING AT 9880
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK20;
CREATE TABLE STOCK20
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_020
            INDEX IN ts_stock_020
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 9881 ENDING AT 10400
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK21;
CREATE TABLE STOCK21
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_021
            INDEX IN ts_stock_021
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 10401 ENDING AT 10920
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK22;
CREATE TABLE STOCK22
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_022
            INDEX IN ts_stock_022
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 10921 ENDING AT 11440
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK23;
CREATE TABLE STOCK23
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_023
            INDEX IN ts_stock_023
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 11441 ENDING AT 11960
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK24;
CREATE TABLE STOCK24
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_024
            INDEX IN ts_stock_024
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 11961 ENDING AT 12480
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK25;
CREATE TABLE STOCK25
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_025
            INDEX IN ts_stock_025
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 12481 ENDING AT 13000
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK26;
CREATE TABLE STOCK26
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_026
            INDEX IN ts_stock_026
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 13001 ENDING AT 13520
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK27;
CREATE TABLE STOCK27
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_027
            INDEX IN ts_stock_027
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 13521 ENDING AT 14040
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK28;
CREATE TABLE STOCK28
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_028
            INDEX IN ts_stock_028
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 14041 ENDING AT 14560
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK29;
CREATE TABLE STOCK29
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_029
            INDEX IN ts_stock_029
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 14561 ENDING AT 15080
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK30;
CREATE TABLE STOCK30
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_030
            INDEX IN ts_stock_030
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 15081 ENDING AT 15600
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK31;
CREATE TABLE STOCK31
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_031
            INDEX IN ts_stock_031
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 15601 ENDING AT 16120
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK32;
CREATE TABLE STOCK32
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_032
            INDEX IN ts_stock_032
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 16121 ENDING AT 16640
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK33;
CREATE TABLE STOCK33
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_033
            INDEX IN ts_stock_033
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 16641 ENDING AT 17160
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK34;
CREATE TABLE STOCK34
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_034
            INDEX IN ts_stock_034
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 17161 ENDING AT 17680
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK35;
CREATE TABLE STOCK35
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_035
            INDEX IN ts_stock_035
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 17681 ENDING AT 18200
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK36;
CREATE TABLE STOCK36
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_036
            INDEX IN ts_stock_036
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 18201 ENDING AT 18720
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK37;
CREATE TABLE STOCK37
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_037
            INDEX IN ts_stock_037
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 18721 ENDING AT 19240
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK38;
CREATE TABLE STOCK38
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_038
            INDEX IN ts_stock_038
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 19241 ENDING AT 19760
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK39;
CREATE TABLE STOCK39
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_039
            INDEX IN ts_stock_039
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 19761 ENDING AT 20280
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK40;
CREATE TABLE STOCK40
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_040
            INDEX IN ts_stock_040
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 20281 ENDING AT 20800
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK41;
CREATE TABLE STOCK41
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_041
            INDEX IN ts_stock_041
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 20801 ENDING AT 21320
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK42;
CREATE TABLE STOCK42
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_042
            INDEX IN ts_stock_042
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 21321 ENDING AT 21840
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK43;
CREATE TABLE STOCK43
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_043
            INDEX IN ts_stock_043
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 21841 ENDING AT 22360
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK44;
CREATE TABLE STOCK44
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_044
            INDEX IN ts_stock_044
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 22361 ENDING AT 22880
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK45;
CREATE TABLE STOCK45
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_045
            INDEX IN ts_stock_045
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 22881 ENDING AT 23400
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK46;
CREATE TABLE STOCK46
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_046
            INDEX IN ts_stock_046
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 23401 ENDING AT 23920
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK47;
CREATE TABLE STOCK47
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_047
            INDEX IN ts_stock_047
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 23921 ENDING AT 24440
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK48;
CREATE TABLE STOCK48
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_048
            INDEX IN ts_stock_048
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 24441 ENDING AT 24960
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK49;
CREATE TABLE STOCK49
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_049
            INDEX IN ts_stock_049
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 24961 ENDING AT 25480
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK50;
CREATE TABLE STOCK50
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_050
            INDEX IN ts_stock_050
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 25481 ENDING AT 26000
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK51;
CREATE TABLE STOCK51
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_051
            INDEX IN ts_stock_051
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 26001 ENDING AT 26520
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK52;
CREATE TABLE STOCK52
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_052
            INDEX IN ts_stock_052
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 26521 ENDING AT 27040
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK53;
CREATE TABLE STOCK53
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_053
            INDEX IN ts_stock_053
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 27041 ENDING AT 27560
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK54;
CREATE TABLE STOCK54
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_054
            INDEX IN ts_stock_054
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 27561 ENDING AT 28080
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK55;
CREATE TABLE STOCK55
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_055
            INDEX IN ts_stock_055
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 28081 ENDING AT 28600
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK56;
CREATE TABLE STOCK56
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_056
            INDEX IN ts_stock_056
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 28601 ENDING AT 29120
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK57;
CREATE TABLE STOCK57
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_057
            INDEX IN ts_stock_057
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 29121 ENDING AT 29640
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK58;
CREATE TABLE STOCK58
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_058
            INDEX IN ts_stock_058
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 29641 ENDING AT 30160
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK59;
CREATE TABLE STOCK59
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_059
            INDEX IN ts_stock_059
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 30161 ENDING AT 30680
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK60;
CREATE TABLE STOCK60
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_060
            INDEX IN ts_stock_060
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 30681 ENDING AT 31200
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK61;
CREATE TABLE STOCK61
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_061
            INDEX IN ts_stock_061
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 31201 ENDING AT 31720
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK62;
CREATE TABLE STOCK62
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_062
            INDEX IN ts_stock_062
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 31721 ENDING AT 32240
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK63;
CREATE TABLE STOCK63
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_063
            INDEX IN ts_stock_063
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 32241 ENDING AT 32760
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK64;
CREATE TABLE STOCK64
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_064
            INDEX IN ts_stock_064
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 32761 ENDING AT 33280
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK65;
CREATE TABLE STOCK65
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_065
            INDEX IN ts_stock_065
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 33281 ENDING AT 33800
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK66;
CREATE TABLE STOCK66
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_066
            INDEX IN ts_stock_066
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 33801 ENDING AT 34320
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK67;
CREATE TABLE STOCK67
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_067
            INDEX IN ts_stock_067
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 34321 ENDING AT 34840
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK68;
CREATE TABLE STOCK68
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_068
            INDEX IN ts_stock_068
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 34841 ENDING AT 35360
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK69;
CREATE TABLE STOCK69
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_069
            INDEX IN ts_stock_069
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 35361 ENDING AT 35880
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK70;
CREATE TABLE STOCK70
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_070
            INDEX IN ts_stock_070
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 35881 ENDING AT 36400
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK71;
CREATE TABLE STOCK71
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_071
            INDEX IN ts_stock_071
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 36401 ENDING AT 36920
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK72;
CREATE TABLE STOCK72
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_072
            INDEX IN ts_stock_072
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 36921 ENDING AT 37440
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK73;
CREATE TABLE STOCK73
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_073
            INDEX IN ts_stock_073
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 37441 ENDING AT 37960
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK74;
CREATE TABLE STOCK74
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_074
            INDEX IN ts_stock_074
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 37961 ENDING AT 38480
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK75;
CREATE TABLE STOCK75
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_075
            INDEX IN ts_stock_075
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 38481 ENDING AT 39000
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK76;
CREATE TABLE STOCK76
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_076
            INDEX IN ts_stock_076
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 39001 ENDING AT 39520
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK77;
CREATE TABLE STOCK77
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_077
            INDEX IN ts_stock_077
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 39521 ENDING AT 40040
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK78;
CREATE TABLE STOCK78
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_078
            INDEX IN ts_stock_078
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 40041 ENDING AT 40560
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK79;
CREATE TABLE STOCK79
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_079
            INDEX IN ts_stock_079
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 40561 ENDING AT 41080
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK80;
CREATE TABLE STOCK80
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_080
            INDEX IN ts_stock_080
```

```
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 41081 ENDING AT 41600
            )
            ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_WAREHOUSE.ddl

```
connect to TPCC in share mode;
DROP TABLE WAREHOUSE1;
CREATE TABLE WAREHOUSE1
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_001
            INDEX IN ts_wh_001
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 1 ENDING AT 520
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE2;
CREATE TABLE WAREHOUSE2
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_002
            INDEX IN ts_wh_002
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 521 ENDING AT 1040
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE3;
CREATE TABLE WAREHOUSE3
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_003
            INDEX IN ts_wh_003
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 1041 ENDING AT 1560
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE4;
CREATE TABLE WAREHOUSE4
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_004
            INDEX IN ts_wh_004
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 1561 ENDING AT 2080
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE5;
CREATE TABLE WAREHOUSE5
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_005
            INDEX IN ts_wh_005
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 2081 ENDING AT 2600
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE6;
CREATE TABLE WAREHOUSE6
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_006
            INDEX IN ts_wh_006
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 2601 ENDING AT 3120
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE7;
CREATE TABLE WAREHOUSE7
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_007
            INDEX IN ts_wh_007
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 3121 ENDING AT 3640
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE8;
CREATE TABLE WAREHOUSE8
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_008
            INDEX IN ts_wh_008
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 3641 ENDING AT 4160
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE9;
CREATE TABLE WAREHOUSE9
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
             W_ID        INTEGER        NOT NULL
            )
            IN ts_wh_009
            INDEX IN ts_wh_009
            ORGANIZE BY KEY SEQUENCE (
             W_ID STARTING FROM 4161 ENDING AT 4680
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE10;
CREATE TABLE WAREHOUSE10
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
             W_CITY      CHAR(20)       NOT NULL,
             W_STATE     CHAR(2)        NOT NULL,
             W_ZIP       CHAR(9)        NOT NULL,
             W_TAX       REAL           NOT NULL,
             W_YTD       DECIMAL(12,2)  NOT NULL,
```

```
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_010
               INDEX IN ts_wh_010
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 4681 ENDING AT 5200
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE11;
CREATE TABLE WAREHOUSE11
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_011
               INDEX IN ts_wh_011
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 5201 ENDING AT 5720
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE12;
CREATE TABLE WAREHOUSE12
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_012
               INDEX IN ts_wh_012
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 5721 ENDING AT 6240
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE13;
CREATE TABLE WAREHOUSE13
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_013
               INDEX IN ts_wh_013
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 6241 ENDING AT 6760
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE14;
CREATE TABLE WAREHOUSE14
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_014
               INDEX IN ts_wh_014
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 6761 ENDING AT 7280
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE15;
CREATE TABLE WAREHOUSE15
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_015
               INDEX IN ts_wh_015
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 7281 ENDING AT 7800
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE16;
CREATE TABLE WAREHOUSE16
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_016
               INDEX IN ts_wh_016
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 7801 ENDING AT 8320
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE17;
CREATE TABLE WAREHOUSE17
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_017
               INDEX IN ts_wh_017
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 8321 ENDING AT 8840
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE18;
CREATE TABLE WAREHOUSE18
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_018
               INDEX IN ts_wh_018
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 8841 ENDING AT 9360
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE19;
CREATE TABLE WAREHOUSE19
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_019
               INDEX IN ts_wh_019
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 9361 ENDING AT 9880
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE20;
CREATE TABLE WAREHOUSE20
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
```

```
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_020
               INDEX IN ts_wh_020
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 9881 ENDING AT 10400
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE21;
CREATE TABLE WAREHOUSE21
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_021
               INDEX IN ts_wh_021
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 10401 ENDING AT 10920
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE22;
CREATE TABLE WAREHOUSE22
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_022
               INDEX IN ts_wh_022
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 10921 ENDING AT 11440
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE23;
CREATE TABLE WAREHOUSE23
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_023
               INDEX IN ts_wh_023
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 11441 ENDING AT 11960
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE24;
CREATE TABLE WAREHOUSE24
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_024
               INDEX IN ts_wh_024
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 11961 ENDING AT 12480
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE25;
CREATE TABLE WAREHOUSE25
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_025
               INDEX IN ts_wh_025
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 12481 ENDING AT 13000
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE26;
CREATE TABLE WAREHOUSE26
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_026
               INDEX IN ts_wh_026
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 13001 ENDING AT 13520
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE27;
CREATE TABLE WAREHOUSE27
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_027
               INDEX IN ts_wh_027
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 13521 ENDING AT 14040
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE28;
CREATE TABLE WAREHOUSE28
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_028
               INDEX IN ts_wh_028
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 14041 ENDING AT 14560
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE29;
CREATE TABLE WAREHOUSE29
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_029
               INDEX IN ts_wh_029
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 14561 ENDING AT 15080
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE30;
CREATE TABLE WAREHOUSE30
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
```

```
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_030
               INDEX IN ts_wh_030
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 15081 ENDING AT 15600
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE31;
CREATE TABLE WAREHOUSE31
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_031
               INDEX IN ts_wh_031
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 15601 ENDING AT 16120
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE32;
CREATE TABLE WAREHOUSE32
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_032
               INDEX IN ts_wh_032
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 16121 ENDING AT 16640
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE33;
CREATE TABLE WAREHOUSE33
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_033
               INDEX IN ts_wh_033
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 16641 ENDING AT 17160
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE34;
CREATE TABLE WAREHOUSE34
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_034
               INDEX IN ts_wh_034
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 17161 ENDING AT 17680
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE35;
CREATE TABLE WAREHOUSE35
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_035
               INDEX IN ts_wh_035
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 17681 ENDING AT 18200
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE36;
CREATE TABLE WAREHOUSE36
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_036
               INDEX IN ts_wh_036
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 18201 ENDING AT 18720
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE37;
CREATE TABLE WAREHOUSE37
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_037
               INDEX IN ts_wh_037
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 18721 ENDING AT 19240
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE38;
CREATE TABLE WAREHOUSE38
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_038
               INDEX IN ts_wh_038
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 19241 ENDING AT 19760
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE39;
CREATE TABLE WAREHOUSE39
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_039
               INDEX IN ts_wh_039
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 19761 ENDING AT 20280
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE40;
CREATE TABLE WAREHOUSE40
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)  NOT NULL,
```

```
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_040
               INDEX IN ts_wh_040
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 20281 ENDING AT 20800
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE41;
CREATE TABLE WAREHOUSE41
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_041
               INDEX IN ts_wh_041
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 20801 ENDING AT 21320
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE42;
CREATE TABLE WAREHOUSE42
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_042
               INDEX IN ts_wh_042
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 21321 ENDING AT 21840
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE43;
CREATE TABLE WAREHOUSE43
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_043
               INDEX IN ts_wh_043
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 21841 ENDING AT 22360
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE44;
CREATE TABLE WAREHOUSE44
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_044
               INDEX IN ts_wh_044
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 22361 ENDING AT 22880
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE45;
CREATE TABLE WAREHOUSE45
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_045
               INDEX IN ts_wh_045
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 22881 ENDING AT 23400
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE46;
CREATE TABLE WAREHOUSE46
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_046
               INDEX IN ts_wh_046
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 23401 ENDING AT 23920
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE47;
CREATE TABLE WAREHOUSE47
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_047
               INDEX IN ts_wh_047
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 23921 ENDING AT 24440
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE48;
CREATE TABLE WAREHOUSE48
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_048
               INDEX IN ts_wh_048
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 24441 ENDING AT 24960
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE49;
CREATE TABLE WAREHOUSE49
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_049
               INDEX IN ts_wh_049
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 24961 ENDING AT 25480
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE50;
CREATE TABLE WAREHOUSE50
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
```

```
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_050
               INDEX IN ts_wh_050
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 25481 ENDING AT 26000
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE51;
CREATE TABLE WAREHOUSE51
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_051
               INDEX IN ts_wh_051
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 26001 ENDING AT 26520
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE52;
CREATE TABLE WAREHOUSE52
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_052
               INDEX IN ts_wh_052
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 26521 ENDING AT 27040
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE53;
CREATE TABLE WAREHOUSE53
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_053
               INDEX IN ts_wh_053
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 27041 ENDING AT 27560
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE54;
CREATE TABLE WAREHOUSE54
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_054
               INDEX IN ts_wh_054
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 27561 ENDING AT 28080
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE55;
CREATE TABLE WAREHOUSE55
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_055
               INDEX IN ts_wh_055
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 28081 ENDING AT 28600
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE56;
CREATE TABLE WAREHOUSE56
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_056
               INDEX IN ts_wh_056
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 28601 ENDING AT 29120
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE57;
CREATE TABLE WAREHOUSE57
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_057
               INDEX IN ts_wh_057
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 29121 ENDING AT 29640
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE58;
CREATE TABLE WAREHOUSE58
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_058
               INDEX IN ts_wh_058
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 29641 ENDING AT 30160
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE59;
CREATE TABLE WAREHOUSE59
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_059
               INDEX IN ts_wh_059
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 30161 ENDING AT 30680
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE60;
CREATE TABLE WAREHOUSE60
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
```

```
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_060
               INDEX IN ts_wh_060
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 30681 ENDING AT 31200
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE61;
CREATE TABLE WAREHOUSE61
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_061
               INDEX IN ts_wh_061
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 31201 ENDING AT 31720
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE62;
CREATE TABLE WAREHOUSE62
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_062
               INDEX IN ts_wh_062
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 31721 ENDING AT 32240
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE63;
CREATE TABLE WAREHOUSE63
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_063
               INDEX IN ts_wh_063
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 32241 ENDING AT 32760
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE64;
CREATE TABLE WAREHOUSE64
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_064
               INDEX IN ts_wh_064
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 32761 ENDING AT 33280
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE65;
CREATE TABLE WAREHOUSE65
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_065
               INDEX IN ts_wh_065
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 33281 ENDING AT 33800
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE66;
CREATE TABLE WAREHOUSE66
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_066
               INDEX IN ts_wh_066
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 33801 ENDING AT 34320
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE67;
CREATE TABLE WAREHOUSE67
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_067
               INDEX IN ts_wh_067
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 34321 ENDING AT 34840
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE68;
CREATE TABLE WAREHOUSE68
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_068
               INDEX IN ts_wh_068
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 34841 ENDING AT 35360
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE69;
CREATE TABLE WAREHOUSE69
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_069
               INDEX IN ts_wh_069
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 35361 ENDING AT 35880
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE70;
CREATE TABLE WAREHOUSE70
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
```

```
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_070
               INDEX IN ts_wh_070
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 35881 ENDING AT 36400
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE71;
CREATE TABLE WAREHOUSE71
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_071
               INDEX IN ts_wh_071
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 36401 ENDING AT 36920
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE72;
CREATE TABLE WAREHOUSE72
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_072
               INDEX IN ts_wh_072
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 36921 ENDING AT 37440
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE73;
CREATE TABLE WAREHOUSE73
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_073
               INDEX IN ts_wh_073
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 37441 ENDING AT 37960
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE74;
CREATE TABLE WAREHOUSE74
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_074
               INDEX IN ts_wh_074
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 37961 ENDING AT 38480
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE75;
CREATE TABLE WAREHOUSE75
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_075
               INDEX IN ts_wh_075
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 38481 ENDING AT 39000
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE76;
CREATE TABLE WAREHOUSE76
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_076
               INDEX IN ts_wh_076
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 39001 ENDING AT 39520
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE77;
CREATE TABLE WAREHOUSE77
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_077
               INDEX IN ts_wh_077
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 39521 ENDING AT 40040
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE78;
CREATE TABLE WAREHOUSE78
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_078
               INDEX IN ts_wh_078
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 40041 ENDING AT 40560
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE79;
CREATE TABLE WAREHOUSE79
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
                W_ID       INTEGER        NOT NULL
               )
               IN ts_wh_079
               INDEX IN ts_wh_079
               ORGANIZE BY KEY SEQUENCE (
                W_ID STARTING FROM 40561 ENDING AT 41080
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE80;
CREATE TABLE WAREHOUSE80
               (
                W_NAME     CHAR(10)       NOT NULL,
                W_STREET_1  CHAR(20)       NOT NULL,
                W_STREET_2  CHAR(20)       NOT NULL,
                W_CITY     CHAR(20)       NOT NULL,
                W_STATE    CHAR(2)        NOT NULL,
                W_ZIP      CHAR(9)        NOT NULL,
                W_TAX      REAL           NOT NULL,
                W_YTD      DECIMAL(12,2)   NOT NULL,
```

```
            W_ID       INTEGER        NOT NULL
           )
           IN ts_wh_080
           INDEX IN ts_wh_080
           ORGANIZE BY KEY SEQUENCE (
            W_ID STARTING FROM 41081 ENDING AT 41600
           )
           ALLOW OVERFLOW;
connect reset;
```

DDL/CRVW_CUSTOMER.ddl

```
connect to TPCC in share mode;
DROP VIEW CUSTOMER;
CREATE VIEW CUSTOMER
           (C_ID,
            C_STATE,
            C_ZIP,
            C_PHONE,
            C_SINCE,
            C_CREDIT_LIM,
            C_MIDDLE,
            C_CREDIT,
            C_DISCOUNT,
            C_DATA,
            C_LAST,
            C_FIRST,
            C_STREET_1,
            C_STREET_2,
            C_CITY,
            C_D_ID,
            C_W_ID,
            C_DELIVERY_CNT,
            C_BALANCE,
            C_YTD_PAYMENT,
            C_PAYMENT_CNT
           ) AS SELECT * FROM CUSTOMER1 UNION ALL
SELECT * FROM CUSTOMER2 UNION ALL
SELECT * FROM CUSTOMER3 UNION ALL
SELECT * FROM CUSTOMER4 UNION ALL
SELECT * FROM CUSTOMER5 UNION ALL
SELECT * FROM CUSTOMER6 UNION ALL
SELECT * FROM CUSTOMER7 UNION ALL
SELECT * FROM CUSTOMER8 UNION ALL
SELECT * FROM CUSTOMER9 UNION ALL
SELECT * FROM CUSTOMER10 UNION ALL
SELECT * FROM CUSTOMER11 UNION ALL
SELECT * FROM CUSTOMER12 UNION ALL
SELECT * FROM CUSTOMER13 UNION ALL
SELECT * FROM CUSTOMER14 UNION ALL
SELECT * FROM CUSTOMER15 UNION ALL
SELECT * FROM CUSTOMER16 UNION ALL
SELECT * FROM CUSTOMER17 UNION ALL
SELECT * FROM CUSTOMER18 UNION ALL
SELECT * FROM CUSTOMER19 UNION ALL
SELECT * FROM CUSTOMER20 UNION ALL
SELECT * FROM CUSTOMER21 UNION ALL
SELECT * FROM CUSTOMER22 UNION ALL
SELECT * FROM CUSTOMER23 UNION ALL
SELECT * FROM CUSTOMER24 UNION ALL
SELECT * FROM CUSTOMER25 UNION ALL
SELECT * FROM CUSTOMER26 UNION ALL
SELECT * FROM CUSTOMER27 UNION ALL
SELECT * FROM CUSTOMER28 UNION ALL
SELECT * FROM CUSTOMER29 UNION ALL
SELECT * FROM CUSTOMER30 UNION ALL
SELECT * FROM CUSTOMER31 UNION ALL
SELECT * FROM CUSTOMER32 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38 UNION ALL
SELECT * FROM CUSTOMER39 UNION ALL
SELECT * FROM CUSTOMER40 UNION ALL
SELECT * FROM CUSTOMER41 UNION ALL
SELECT * FROM CUSTOMER42 UNION ALL
SELECT * FROM CUSTOMER43 UNION ALL
SELECT * FROM CUSTOMER44 UNION ALL
SELECT * FROM CUSTOMER45 UNION ALL
SELECT * FROM CUSTOMER46 UNION ALL
SELECT * FROM CUSTOMER47 UNION ALL
SELECT * FROM CUSTOMER48 UNION ALL
SELECT * FROM CUSTOMER49 UNION ALL
SELECT * FROM CUSTOMER50 UNION ALL
SELECT * FROM CUSTOMER51 UNION ALL
SELECT * FROM CUSTOMER52 UNION ALL
SELECT * FROM CUSTOMER53 UNION ALL
SELECT * FROM CUSTOMER54 UNION ALL
SELECT * FROM CUSTOMER55 UNION ALL
SELECT * FROM CUSTOMER56 UNION ALL
SELECT * FROM CUSTOMER57 UNION ALL
SELECT * FROM CUSTOMER58 UNION ALL
SELECT * FROM CUSTOMER59 UNION ALL
SELECT * FROM CUSTOMER60 UNION ALL
SELECT * FROM CUSTOMER61 UNION ALL
SELECT * FROM CUSTOMER62 UNION ALL
SELECT * FROM CUSTOMER63 UNION ALL
SELECT * FROM CUSTOMER64 UNION ALL
SELECT * FROM CUSTOMER65 UNION ALL
SELECT * FROM CUSTOMER66 UNION ALL
SELECT * FROM CUSTOMER67 UNION ALL
SELECT * FROM CUSTOMER68 UNION ALL
SELECT * FROM CUSTOMER69 UNION ALL
SELECT * FROM CUSTOMER70 UNION ALL
SELECT * FROM CUSTOMER71 UNION ALL
SELECT * FROM CUSTOMER72 UNION ALL
SELECT * FROM CUSTOMER73 UNION ALL
SELECT * FROM CUSTOMER74 UNION ALL
SELECT * FROM CUSTOMER75 UNION ALL
SELECT * FROM CUSTOMER76 UNION ALL
SELECT * FROM CUSTOMER77 UNION ALL
SELECT * FROM CUSTOMER78 UNION ALL
SELECT * FROM CUSTOMER79 UNION ALL
SELECT * FROM CUSTOMER80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_DISTRICT.ddl

```
connect to TPCC in share mode;
DROP VIEW DISTRICT;
CREATE VIEW DISTRICT
           (D_NEXT_O_ID,
            D_TAX,
            D_YTD,
            D_NAME,
            D_STREET_1,
            D_STREET_2,
            D_CITY,
            D_STATE,
            D_ZIP,
            D_ID,
            D_W_ID
           ) AS SELECT * FROM DISTRICT1 UNION ALL
SELECT * FROM DISTRICT2 UNION ALL
SELECT * FROM DISTRICT3 UNION ALL
SELECT * FROM DISTRICT4 UNION ALL
SELECT * FROM DISTRICT5 UNION ALL
SELECT * FROM DISTRICT6 UNION ALL
SELECT * FROM DISTRICT7 UNION ALL
SELECT * FROM DISTRICT8 UNION ALL
SELECT * FROM DISTRICT9 UNION ALL
SELECT * FROM DISTRICT10 UNION ALL
SELECT * FROM DISTRICT11 UNION ALL
SELECT * FROM DISTRICT12 UNION ALL
SELECT * FROM DISTRICT13 UNION ALL
SELECT * FROM DISTRICT14 UNION ALL
SELECT * FROM DISTRICT15 UNION ALL
SELECT * FROM DISTRICT16 UNION ALL
SELECT * FROM DISTRICT17 UNION ALL
SELECT * FROM DISTRICT18 UNION ALL
SELECT * FROM DISTRICT19 UNION ALL
SELECT * FROM DISTRICT20 UNION ALL
SELECT * FROM DISTRICT21 UNION ALL
SELECT * FROM DISTRICT22 UNION ALL
SELECT * FROM DISTRICT23 UNION ALL
SELECT * FROM DISTRICT24 UNION ALL
SELECT * FROM DISTRICT25 UNION ALL
SELECT * FROM DISTRICT26 UNION ALL
SELECT * FROM DISTRICT27 UNION ALL
SELECT * FROM DISTRICT28 UNION ALL
SELECT * FROM DISTRICT29 UNION ALL
SELECT * FROM DISTRICT30 UNION ALL
SELECT * FROM DISTRICT31 UNION ALL
SELECT * FROM DISTRICT32 UNION ALL
SELECT * FROM DISTRICT33 UNION ALL
SELECT * FROM DISTRICT34 UNION ALL
SELECT * FROM DISTRICT35 UNION ALL
SELECT * FROM DISTRICT36 UNION ALL
SELECT * FROM DISTRICT37 UNION ALL
SELECT * FROM DISTRICT38 UNION ALL
SELECT * FROM DISTRICT39 UNION ALL
SELECT * FROM DISTRICT40 UNION ALL
SELECT * FROM DISTRICT41 UNION ALL
SELECT * FROM DISTRICT42 UNION ALL
SELECT * FROM DISTRICT43 UNION ALL
SELECT * FROM DISTRICT44 UNION ALL
SELECT * FROM DISTRICT45 UNION ALL
SELECT * FROM DISTRICT46 UNION ALL
SELECT * FROM DISTRICT47 UNION ALL
SELECT * FROM DISTRICT48 UNION ALL
SELECT * FROM DISTRICT49 UNION ALL
SELECT * FROM DISTRICT50 UNION ALL
SELECT * FROM DISTRICT51 UNION ALL
SELECT * FROM DISTRICT52 UNION ALL
SELECT * FROM DISTRICT53 UNION ALL
SELECT * FROM DISTRICT54 UNION ALL
SELECT * FROM DISTRICT55 UNION ALL
SELECT * FROM DISTRICT56 UNION ALL
SELECT * FROM DISTRICT57 UNION ALL
SELECT * FROM DISTRICT58 UNION ALL
SELECT * FROM DISTRICT59 UNION ALL
SELECT * FROM DISTRICT60 UNION ALL
SELECT * FROM DISTRICT61 UNION ALL
SELECT * FROM DISTRICT62 UNION ALL
SELECT * FROM DISTRICT63 UNION ALL
SELECT * FROM DISTRICT64 UNION ALL
SELECT * FROM DISTRICT65 UNION ALL
SELECT * FROM DISTRICT66 UNION ALL
SELECT * FROM DISTRICT67 UNION ALL
SELECT * FROM DISTRICT68 UNION ALL
SELECT * FROM DISTRICT69 UNION ALL
SELECT * FROM DISTRICT70 UNION ALL
SELECT * FROM DISTRICT71 UNION ALL
SELECT * FROM DISTRICT72 UNION ALL
SELECT * FROM DISTRICT73 UNION ALL
```

```
SELECT * FROM DISTRICT74 UNION ALL
SELECT * FROM DISTRICT75 UNION ALL
SELECT * FROM DISTRICT76 UNION ALL
SELECT * FROM DISTRICT77 UNION ALL
SELECT * FROM DISTRICT78 UNION ALL
SELECT * FROM DISTRICT79 UNION ALL
SELECT * FROM DISTRICT80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_HISTORY.ddl

```
connect to TPCC in share mode;
DROP VIEW HISTORY;
CREATE VIEW HISTORY
            (H_C_ID,
             H_C_D_ID,
             H_C_W_ID,
             H_D_ID,
             H_W_ID,
             H_DATE,
             H_AMOUNT,
             H_DATA
            ) AS SELECT * FROM HISTORY1 UNION ALL
SELECT * FROM HISTORY2 UNION ALL
SELECT * FROM HISTORY3 UNION ALL
SELECT * FROM HISTORY4 UNION ALL
SELECT * FROM HISTORY5 UNION ALL
SELECT * FROM HISTORY6 UNION ALL
SELECT * FROM HISTORY7 UNION ALL
SELECT * FROM HISTORY8 UNION ALL
SELECT * FROM HISTORY9 UNION ALL
SELECT * FROM HISTORY10 UNION ALL
SELECT * FROM HISTORY11 UNION ALL
SELECT * FROM HISTORY12 UNION ALL
SELECT * FROM HISTORY13 UNION ALL
SELECT * FROM HISTORY14 UNION ALL
SELECT * FROM HISTORY15 UNION ALL
SELECT * FROM HISTORY16 UNION ALL
SELECT * FROM HISTORY17 UNION ALL
SELECT * FROM HISTORY18 UNION ALL
SELECT * FROM HISTORY19 UNION ALL
SELECT * FROM HISTORY20 UNION ALL
SELECT * FROM HISTORY21 UNION ALL
SELECT * FROM HISTORY22 UNION ALL
SELECT * FROM HISTORY23 UNION ALL
SELECT * FROM HISTORY24 UNION ALL
SELECT * FROM HISTORY25 UNION ALL
SELECT * FROM HISTORY26 UNION ALL
SELECT * FROM HISTORY27 UNION ALL
SELECT * FROM HISTORY28 UNION ALL
SELECT * FROM HISTORY29 UNION ALL
SELECT * FROM HISTORY30 UNION ALL
SELECT * FROM HISTORY31 UNION ALL
SELECT * FROM HISTORY32 UNION ALL
SELECT * FROM HISTORY33 UNION ALL
SELECT * FROM HISTORY34 UNION ALL
SELECT * FROM HISTORY35 UNION ALL
SELECT * FROM HISTORY36 UNION ALL
SELECT * FROM HISTORY37 UNION ALL
SELECT * FROM HISTORY38 UNION ALL
SELECT * FROM HISTORY39 UNION ALL
SELECT * FROM HISTORY40 UNION ALL
SELECT * FROM HISTORY41 UNION ALL
SELECT * FROM HISTORY42 UNION ALL
SELECT * FROM HISTORY43 UNION ALL
SELECT * FROM HISTORY44 UNION ALL
SELECT * FROM HISTORY45 UNION ALL
SELECT * FROM HISTORY46 UNION ALL
SELECT * FROM HISTORY47 UNION ALL
SELECT * FROM HISTORY48 UNION ALL
SELECT * FROM HISTORY49 UNION ALL
SELECT * FROM HISTORY50 UNION ALL
SELECT * FROM HISTORY51 UNION ALL
SELECT * FROM HISTORY52 UNION ALL
SELECT * FROM HISTORY53 UNION ALL
SELECT * FROM HISTORY54 UNION ALL
SELECT * FROM HISTORY55 UNION ALL
SELECT * FROM HISTORY56 UNION ALL
SELECT * FROM HISTORY57 UNION ALL
SELECT * FROM HISTORY58 UNION ALL
SELECT * FROM HISTORY59 UNION ALL
SELECT * FROM HISTORY60 UNION ALL
SELECT * FROM HISTORY61 UNION ALL
SELECT * FROM HISTORY62 UNION ALL
SELECT * FROM HISTORY63 UNION ALL
SELECT * FROM HISTORY64 UNION ALL
SELECT * FROM HISTORY65 UNION ALL
SELECT * FROM HISTORY66 UNION ALL
SELECT * FROM HISTORY67 UNION ALL
SELECT * FROM HISTORY68 UNION ALL
SELECT * FROM HISTORY69 UNION ALL
SELECT * FROM HISTORY70 UNION ALL
SELECT * FROM HISTORY71 UNION ALL
SELECT * FROM HISTORY72 UNION ALL
SELECT * FROM HISTORY73 UNION ALL
SELECT * FROM HISTORY74 UNION ALL
SELECT * FROM HISTORY75 UNION ALL
SELECT * FROM HISTORY76 UNION ALL
SELECT * FROM HISTORY77 UNION ALL
SELECT * FROM HISTORY78 UNION ALL
SELECT * FROM HISTORY79 UNION ALL
SELECT * FROM HISTORY80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_NEW_ORDER.ddl

```
connect to TPCC in share mode;
DROP VIEW NEW_ORDER;
CREATE VIEW NEW_ORDER
            (NO_O_ID,
             NO_D_ID,
             NO_W_ID
            ) AS SELECT * FROM NEW_ORDERA1 UNION ALL
SELECT * FROM NEW_ORDERA2 UNION ALL
SELECT * FROM NEW_ORDERA3 UNION ALL
SELECT * FROM NEW_ORDERA4 UNION ALL
SELECT * FROM NEW_ORDERA5 UNION ALL
SELECT * FROM NEW_ORDERA6 UNION ALL
SELECT * FROM NEW_ORDERA7 UNION ALL
SELECT * FROM NEW_ORDERA8 UNION ALL
SELECT * FROM NEW_ORDERA9 UNION ALL
SELECT * FROM NEW_ORDERA10 UNION ALL
SELECT * FROM NEW_ORDERA11 UNION ALL
SELECT * FROM NEW_ORDERA12 UNION ALL
SELECT * FROM NEW_ORDERA13 UNION ALL
SELECT * FROM NEW_ORDERA14 UNION ALL
SELECT * FROM NEW_ORDERA15 UNION ALL
SELECT * FROM NEW_ORDERA16 UNION ALL
SELECT * FROM NEW_ORDERA17 UNION ALL
SELECT * FROM NEW_ORDERA18 UNION ALL
SELECT * FROM NEW_ORDERA19 UNION ALL
SELECT * FROM NEW_ORDERA20 UNION ALL
SELECT * FROM NEW_ORDERA21 UNION ALL
SELECT * FROM NEW_ORDERA22 UNION ALL
SELECT * FROM NEW_ORDERA23 UNION ALL
SELECT * FROM NEW_ORDERA24 UNION ALL
SELECT * FROM NEW_ORDERA25 UNION ALL
SELECT * FROM NEW_ORDERA26 UNION ALL
SELECT * FROM NEW_ORDERA27 UNION ALL
SELECT * FROM NEW_ORDERA28 UNION ALL
SELECT * FROM NEW_ORDERA29 UNION ALL
SELECT * FROM NEW_ORDERA30 UNION ALL
SELECT * FROM NEW_ORDERA31 UNION ALL
SELECT * FROM NEW_ORDERA32 UNION ALL
SELECT * FROM NEW_ORDERA33 UNION ALL
SELECT * FROM NEW_ORDERA34 UNION ALL
SELECT * FROM NEW_ORDERA35 UNION ALL
SELECT * FROM NEW_ORDERA36 UNION ALL
SELECT * FROM NEW_ORDERA37 UNION ALL
SELECT * FROM NEW_ORDERA38 UNION ALL
SELECT * FROM NEW_ORDERA39 UNION ALL
SELECT * FROM NEW_ORDERA40 UNION ALL
SELECT * FROM NEW_ORDERA41 UNION ALL
SELECT * FROM NEW_ORDERA42 UNION ALL
SELECT * FROM NEW_ORDERA43 UNION ALL
SELECT * FROM NEW_ORDERA44 UNION ALL
SELECT * FROM NEW_ORDERA45 UNION ALL
SELECT * FROM NEW_ORDERA46 UNION ALL
SELECT * FROM NEW_ORDERA47 UNION ALL
SELECT * FROM NEW_ORDERA48 UNION ALL
SELECT * FROM NEW_ORDERA49 UNION ALL
SELECT * FROM NEW_ORDERA50 UNION ALL
SELECT * FROM NEW_ORDERA51 UNION ALL
SELECT * FROM NEW_ORDERA52 UNION ALL
SELECT * FROM NEW_ORDERA53 UNION ALL
SELECT * FROM NEW_ORDERA54 UNION ALL
SELECT * FROM NEW_ORDERA55 UNION ALL
SELECT * FROM NEW_ORDERA56 UNION ALL
SELECT * FROM NEW_ORDERA57 UNION ALL
SELECT * FROM NEW_ORDERA58 UNION ALL
SELECT * FROM NEW_ORDERA59 UNION ALL
SELECT * FROM NEW_ORDERA60 UNION ALL
SELECT * FROM NEW_ORDERA61 UNION ALL
SELECT * FROM NEW_ORDERA62 UNION ALL
SELECT * FROM NEW_ORDERA63 UNION ALL
SELECT * FROM NEW_ORDERA64 UNION ALL
SELECT * FROM NEW_ORDERA65 UNION ALL
SELECT * FROM NEW_ORDERA66 UNION ALL
SELECT * FROM NEW_ORDERA67 UNION ALL
SELECT * FROM NEW_ORDERA68 UNION ALL
SELECT * FROM NEW_ORDERA69 UNION ALL
SELECT * FROM NEW_ORDERA70 UNION ALL
SELECT * FROM NEW_ORDERA71 UNION ALL
SELECT * FROM NEW_ORDERA72 UNION ALL
SELECT * FROM NEW_ORDERA73 UNION ALL
SELECT * FROM NEW_ORDERA74 UNION ALL
SELECT * FROM NEW_ORDERA75 UNION ALL
SELECT * FROM NEW_ORDERA76 UNION ALL
SELECT * FROM NEW_ORDERA77 UNION ALL
SELECT * FROM NEW_ORDERA78 UNION ALL
SELECT * FROM NEW_ORDERA79 UNION ALL
SELECT * FROM NEW_ORDERA80 UNION ALL
SELECT * FROM NEW_ORDERB1 UNION ALL
SELECT * FROM NEW_ORDERB2 UNION ALL
SELECT * FROM NEW_ORDERB3 UNION ALL
SELECT * FROM NEW_ORDERB4 UNION ALL
SELECT * FROM NEW_ORDERB5 UNION ALL
SELECT * FROM NEW_ORDERB6 UNION ALL
SELECT * FROM NEW_ORDERB7 UNION ALL
SELECT * FROM NEW_ORDERB8 UNION ALL
SELECT * FROM NEW_ORDERB9 UNION ALL
SELECT * FROM NEW_ORDERB10 UNION ALL
SELECT * FROM NEW_ORDERB11 UNION ALL
SELECT * FROM NEW_ORDERB12 UNION ALL
```

```
SELECT * FROM NEW_ORDERB13 UNION ALL
SELECT * FROM NEW_ORDERB14 UNION ALL
SELECT * FROM NEW_ORDERB15 UNION ALL
SELECT * FROM NEW_ORDERB16 UNION ALL
SELECT * FROM NEW_ORDERB17 UNION ALL
SELECT * FROM NEW_ORDERB18 UNION ALL
SELECT * FROM NEW_ORDERB19 UNION ALL
SELECT * FROM NEW_ORDERB20 UNION ALL
SELECT * FROM NEW_ORDERB21 UNION ALL
SELECT * FROM NEW_ORDERB22 UNION ALL
SELECT * FROM NEW_ORDERB23 UNION ALL
SELECT * FROM NEW_ORDERB24 UNION ALL
SELECT * FROM NEW_ORDERB25 UNION ALL
SELECT * FROM NEW_ORDERB26 UNION ALL
SELECT * FROM NEW_ORDERB27 UNION ALL
SELECT * FROM NEW_ORDERB28 UNION ALL
SELECT * FROM NEW_ORDERB29 UNION ALL
SELECT * FROM NEW_ORDERB30 UNION ALL
SELECT * FROM NEW_ORDERB31 UNION ALL
SELECT * FROM NEW_ORDERB32 UNION ALL
SELECT * FROM NEW_ORDERB33 UNION ALL
SELECT * FROM NEW_ORDERB34 UNION ALL
SELECT * FROM NEW_ORDERB35 UNION ALL
SELECT * FROM NEW_ORDERB36 UNION ALL
SELECT * FROM NEW_ORDERB37 UNION ALL
SELECT * FROM NEW_ORDERB38 UNION ALL
SELECT * FROM NEW_ORDERB39 UNION ALL
SELECT * FROM NEW_ORDERB40 UNION ALL
SELECT * FROM NEW_ORDERB41 UNION ALL
SELECT * FROM NEW_ORDERB42 UNION ALL
SELECT * FROM NEW_ORDERB43 UNION ALL
SELECT * FROM NEW_ORDERB44 UNION ALL
SELECT * FROM NEW_ORDERB45 UNION ALL
SELECT * FROM NEW_ORDERB46 UNION ALL
SELECT * FROM NEW_ORDERB47 UNION ALL
SELECT * FROM NEW_ORDERB48 UNION ALL
SELECT * FROM NEW_ORDERB49 UNION ALL
SELECT * FROM NEW_ORDERB50 UNION ALL
SELECT * FROM NEW_ORDERB51 UNION ALL
SELECT * FROM NEW_ORDERB52 UNION ALL
SELECT * FROM NEW_ORDERB53 UNION ALL
SELECT * FROM NEW_ORDERB54 UNION ALL
SELECT * FROM NEW_ORDERB55 UNION ALL
SELECT * FROM NEW_ORDERB56 UNION ALL
SELECT * FROM NEW_ORDERB57 UNION ALL
SELECT * FROM NEW_ORDERB58 UNION ALL
SELECT * FROM NEW_ORDERB59 UNION ALL
SELECT * FROM NEW_ORDERB60 UNION ALL
SELECT * FROM NEW_ORDERB61 UNION ALL
SELECT * FROM NEW_ORDERB62 UNION ALL
SELECT * FROM NEW_ORDERB63 UNION ALL
SELECT * FROM NEW_ORDERB64 UNION ALL
SELECT * FROM NEW_ORDERB65 UNION ALL
SELECT * FROM NEW_ORDERB66 UNION ALL
SELECT * FROM NEW_ORDERB67 UNION ALL
SELECT * FROM NEW_ORDERB68 UNION ALL
SELECT * FROM NEW_ORDERB69 UNION ALL
SELECT * FROM NEW_ORDERB70 UNION ALL
SELECT * FROM NEW_ORDERB71 UNION ALL
SELECT * FROM NEW_ORDERB72 UNION ALL
SELECT * FROM NEW_ORDERB73 UNION ALL
SELECT * FROM NEW_ORDERB74 UNION ALL
SELECT * FROM NEW_ORDERB75 UNION ALL
SELECT * FROM NEW_ORDERB76 UNION ALL
SELECT * FROM NEW_ORDERB77 UNION ALL
SELECT * FROM NEW_ORDERB78 UNION ALL
SELECT * FROM NEW_ORDERB79 UNION ALL
SELECT * FROM NEW_ORDERB80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_ORDERS.ddl

```
connect to TPCC in share mode;
DROP VIEW ORDERS;
CREATE VIEW ORDERS
            (O_C_ID,
             O_ENTRY_D,
             O_CARRIER_ID,
             O_OL_CNT,
             O_ALL_LOCAL,
             O_ID,
             O_W_ID,
             O_D_ID
            ) AS SELECT * FROM ORDERS1 UNION ALL
SELECT * FROM ORDERS2 UNION ALL
SELECT * FROM ORDERS3 UNION ALL
SELECT * FROM ORDERS4 UNION ALL
SELECT * FROM ORDERS5 UNION ALL
SELECT * FROM ORDERS6 UNION ALL
SELECT * FROM ORDERS7 UNION ALL
SELECT * FROM ORDERS8 UNION ALL
SELECT * FROM ORDERS9 UNION ALL
SELECT * FROM ORDERS10 UNION ALL
SELECT * FROM ORDERS11 UNION ALL
SELECT * FROM ORDERS12 UNION ALL
SELECT * FROM ORDERS13 UNION ALL
SELECT * FROM ORDERS14 UNION ALL
SELECT * FROM ORDERS15 UNION ALL
SELECT * FROM ORDERS16 UNION ALL
SELECT * FROM ORDERS17 UNION ALL
SELECT * FROM ORDERS18 UNION ALL
SELECT * FROM ORDERS19 UNION ALL
SELECT * FROM ORDERS20 UNION ALL
SELECT * FROM ORDERS21 UNION ALL
SELECT * FROM ORDERS22 UNION ALL
SELECT * FROM ORDERS23 UNION ALL
SELECT * FROM ORDERS24 UNION ALL
SELECT * FROM ORDERS25 UNION ALL
SELECT * FROM ORDERS26 UNION ALL
SELECT * FROM ORDERS27 UNION ALL
SELECT * FROM ORDERS28 UNION ALL
SELECT * FROM ORDERS29 UNION ALL
SELECT * FROM ORDERS30 UNION ALL
SELECT * FROM ORDERS31 UNION ALL
SELECT * FROM ORDERS32 UNION ALL
SELECT * FROM ORDERS33 UNION ALL
SELECT * FROM ORDERS34 UNION ALL
SELECT * FROM ORDERS35 UNION ALL
SELECT * FROM ORDERS36 UNION ALL
SELECT * FROM ORDERS37 UNION ALL
SELECT * FROM ORDERS38 UNION ALL
SELECT * FROM ORDERS39 UNION ALL
SELECT * FROM ORDERS40 UNION ALL
SELECT * FROM ORDERS41 UNION ALL
SELECT * FROM ORDERS42 UNION ALL
SELECT * FROM ORDERS43 UNION ALL
SELECT * FROM ORDERS44 UNION ALL
SELECT * FROM ORDERS45 UNION ALL
SELECT * FROM ORDERS46 UNION ALL
SELECT * FROM ORDERS47 UNION ALL
SELECT * FROM ORDERS48 UNION ALL
SELECT * FROM ORDERS49 UNION ALL
SELECT * FROM ORDERS50 UNION ALL
SELECT * FROM ORDERS51 UNION ALL
SELECT * FROM ORDERS52 UNION ALL
SELECT * FROM ORDERS53 UNION ALL
SELECT * FROM ORDERS54 UNION ALL
SELECT * FROM ORDERS55 UNION ALL
SELECT * FROM ORDERS56 UNION ALL
SELECT * FROM ORDERS57 UNION ALL
SELECT * FROM ORDERS58 UNION ALL
SELECT * FROM ORDERS59 UNION ALL
SELECT * FROM ORDERS60 UNION ALL
SELECT * FROM ORDERS61 UNION ALL
SELECT * FROM ORDERS62 UNION ALL
SELECT * FROM ORDERS63 UNION ALL
SELECT * FROM ORDERS64 UNION ALL
SELECT * FROM ORDERS65 UNION ALL
SELECT * FROM ORDERS66 UNION ALL
SELECT * FROM ORDERS67 UNION ALL
SELECT * FROM ORDERS68 UNION ALL
SELECT * FROM ORDERS69 UNION ALL
SELECT * FROM ORDERS70 UNION ALL
SELECT * FROM ORDERS71 UNION ALL
SELECT * FROM ORDERS72 UNION ALL
SELECT * FROM ORDERS73 UNION ALL
SELECT * FROM ORDERS74 UNION ALL
SELECT * FROM ORDERS75 UNION ALL
SELECT * FROM ORDERS76 UNION ALL
SELECT * FROM ORDERS77 UNION ALL
SELECT * FROM ORDERS78 UNION ALL
SELECT * FROM ORDERS79 UNION ALL
SELECT * FROM ORDERS80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_ORDER_LINE.ddl

```
connect to TPCC in share mode;
DROP VIEW ORDER_LINE;
CREATE VIEW ORDER_LINE
            (OL_DELIVERY_D,
             OL_AMOUNT,
             OL_I_ID,
             OL_SUPPLY_W_ID,
             OL_QUANTITY,
             OL_DIST_INFO,
             OL_O_ID,
             OL_D_ID,
             OL_W_ID,
             OL_NUMBER
            ) AS SELECT * FROM ORDER_LINE1 UNION ALL
SELECT * FROM ORDER_LINE2 UNION ALL
SELECT * FROM ORDER_LINE3 UNION ALL
SELECT * FROM ORDER_LINE4 UNION ALL
SELECT * FROM ORDER_LINE5 UNION ALL
SELECT * FROM ORDER_LINE6 UNION ALL
SELECT * FROM ORDER_LINE7 UNION ALL
SELECT * FROM ORDER_LINE8 UNION ALL
SELECT * FROM ORDER_LINE9 UNION ALL
SELECT * FROM ORDER_LINE10 UNION ALL
SELECT * FROM ORDER_LINE11 UNION ALL
SELECT * FROM ORDER_LINE12 UNION ALL
SELECT * FROM ORDER_LINE13 UNION ALL
SELECT * FROM ORDER_LINE14 UNION ALL
SELECT * FROM ORDER_LINE15 UNION ALL
SELECT * FROM ORDER_LINE16 UNION ALL
SELECT * FROM ORDER_LINE17 UNION ALL
SELECT * FROM ORDER_LINE18 UNION ALL
SELECT * FROM ORDER_LINE19 UNION ALL
SELECT * FROM ORDER_LINE20 UNION ALL
SELECT * FROM ORDER_LINE21 UNION ALL
SELECT * FROM ORDER_LINE22 UNION ALL
SELECT * FROM ORDER_LINE23 UNION ALL
SELECT * FROM ORDER_LINE24 UNION ALL
SELECT * FROM ORDER_LINE25 UNION ALL
SELECT * FROM ORDER_LINE26 UNION ALL
```

```
SELECT * FROM ORDER_LINE27 UNION ALL
SELECT * FROM ORDER_LINE28 UNION ALL
SELECT * FROM ORDER_LINE29 UNION ALL
SELECT * FROM ORDER_LINE30 UNION ALL
SELECT * FROM ORDER_LINE31 UNION ALL
SELECT * FROM ORDER_LINE32 UNION ALL
SELECT * FROM ORDER_LINE33 UNION ALL
SELECT * FROM ORDER_LINE34 UNION ALL
SELECT * FROM ORDER_LINE35 UNION ALL
SELECT * FROM ORDER_LINE36 UNION ALL
SELECT * FROM ORDER_LINE37 UNION ALL
SELECT * FROM ORDER_LINE38 UNION ALL
SELECT * FROM ORDER_LINE39 UNION ALL
SELECT * FROM ORDER_LINE40 UNION ALL
SELECT * FROM ORDER_LINE41 UNION ALL
SELECT * FROM ORDER_LINE42 UNION ALL
SELECT * FROM ORDER_LINE43 UNION ALL
SELECT * FROM ORDER_LINE44 UNION ALL
SELECT * FROM ORDER_LINE45 UNION ALL
SELECT * FROM ORDER_LINE46 UNION ALL
SELECT * FROM ORDER_LINE47 UNION ALL
SELECT * FROM ORDER_LINE48 UNION ALL
SELECT * FROM ORDER_LINE49 UNION ALL
SELECT * FROM ORDER_LINE50 UNION ALL
SELECT * FROM ORDER_LINE51 UNION ALL
SELECT * FROM ORDER_LINE52 UNION ALL
SELECT * FROM ORDER_LINE53 UNION ALL
SELECT * FROM ORDER_LINE54 UNION ALL
SELECT * FROM ORDER_LINE55 UNION ALL
SELECT * FROM ORDER_LINE56 UNION ALL
SELECT * FROM ORDER_LINE57 UNION ALL
SELECT * FROM ORDER_LINE58 UNION ALL
SELECT * FROM ORDER_LINE59 UNION ALL
SELECT * FROM ORDER_LINE60 UNION ALL
SELECT * FROM ORDER_LINE61 UNION ALL
SELECT * FROM ORDER_LINE62 UNION ALL
SELECT * FROM ORDER_LINE63 UNION ALL
SELECT * FROM ORDER_LINE64 UNION ALL
SELECT * FROM ORDER_LINE65 UNION ALL
SELECT * FROM ORDER_LINE66 UNION ALL
SELECT * FROM ORDER_LINE67 UNION ALL
SELECT * FROM ORDER_LINE68 UNION ALL
SELECT * FROM ORDER_LINE69 UNION ALL
SELECT * FROM ORDER_LINE70 UNION ALL
SELECT * FROM ORDER_LINE71 UNION ALL
SELECT * FROM ORDER_LINE72 UNION ALL
SELECT * FROM ORDER_LINE73 UNION ALL
SELECT * FROM ORDER_LINE74 UNION ALL
SELECT * FROM ORDER_LINE75 UNION ALL
SELECT * FROM ORDER_LINE76 UNION ALL
SELECT * FROM ORDER_LINE77 UNION ALL
SELECT * FROM ORDER_LINE78 UNION ALL
SELECT * FROM ORDER_LINE79 UNION ALL
SELECT * FROM ORDER_LINE80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_STOCK.ddl

```
connect to TPCC in share mode;
DROP VIEW STOCK;
CREATE VIEW STOCK
            (S_REMOTE_CNT,
             S_QUANTITY,
             S_ORDER_CNT,
             S_YTD,
             S_DATA,
             S_DIST_01,
             S_DIST_02,
             S_DIST_03,
             S_DIST_04,
             S_DIST_05,
             S_DIST_06,
             S_DIST_07,
             S_DIST_08,
             S_DIST_09,
             S_DIST_10,
             S_I_ID,
             S_W_ID
            ) AS SELECT * FROM STOCK1 UNION ALL
SELECT * FROM STOCK2 UNION ALL
SELECT * FROM STOCK3 UNION ALL
SELECT * FROM STOCK4 UNION ALL
SELECT * FROM STOCK5 UNION ALL
SELECT * FROM STOCK6 UNION ALL
SELECT * FROM STOCK7 UNION ALL
SELECT * FROM STOCK8 UNION ALL
SELECT * FROM STOCK9 UNION ALL
SELECT * FROM STOCK10 UNION ALL
SELECT * FROM STOCK11 UNION ALL
SELECT * FROM STOCK12 UNION ALL
SELECT * FROM STOCK13 UNION ALL
SELECT * FROM STOCK14 UNION ALL
SELECT * FROM STOCK15 UNION ALL
SELECT * FROM STOCK16 UNION ALL
SELECT * FROM STOCK17 UNION ALL
SELECT * FROM STOCK18 UNION ALL
SELECT * FROM STOCK19 UNION ALL
SELECT * FROM STOCK20 UNION ALL
SELECT * FROM STOCK21 UNION ALL
SELECT * FROM STOCK22 UNION ALL
SELECT * FROM STOCK23 UNION ALL
SELECT * FROM STOCK24 UNION ALL
SELECT * FROM STOCK25 UNION ALL
SELECT * FROM STOCK26 UNION ALL
SELECT * FROM STOCK27 UNION ALL
SELECT * FROM STOCK28 UNION ALL
SELECT * FROM STOCK29 UNION ALL
SELECT * FROM STOCK30 UNION ALL
SELECT * FROM STOCK31 UNION ALL
SELECT * FROM STOCK32 UNION ALL
SELECT * FROM STOCK33 UNION ALL
SELECT * FROM STOCK34 UNION ALL
SELECT * FROM STOCK35 UNION ALL
SELECT * FROM STOCK36 UNION ALL
SELECT * FROM STOCK37 UNION ALL
SELECT * FROM STOCK38 UNION ALL
SELECT * FROM STOCK39 UNION ALL
SELECT * FROM STOCK40 UNION ALL
SELECT * FROM STOCK41 UNION ALL
SELECT * FROM STOCK42 UNION ALL
SELECT * FROM STOCK43 UNION ALL
SELECT * FROM STOCK44 UNION ALL
SELECT * FROM STOCK45 UNION ALL
SELECT * FROM STOCK46 UNION ALL
SELECT * FROM STOCK47 UNION ALL
SELECT * FROM STOCK48 UNION ALL
SELECT * FROM STOCK49 UNION ALL
SELECT * FROM STOCK50 UNION ALL
SELECT * FROM STOCK51 UNION ALL
SELECT * FROM STOCK52 UNION ALL
SELECT * FROM STOCK53 UNION ALL
SELECT * FROM STOCK54 UNION ALL
SELECT * FROM STOCK55 UNION ALL
SELECT * FROM STOCK56 UNION ALL
SELECT * FROM STOCK57 UNION ALL
SELECT * FROM STOCK58 UNION ALL
SELECT * FROM STOCK59 UNION ALL
SELECT * FROM STOCK60 UNION ALL
SELECT * FROM STOCK61 UNION ALL
SELECT * FROM STOCK62 UNION ALL
SELECT * FROM STOCK63 UNION ALL
SELECT * FROM STOCK64 UNION ALL
SELECT * FROM STOCK65 UNION ALL
SELECT * FROM STOCK66 UNION ALL
SELECT * FROM STOCK67 UNION ALL
SELECT * FROM STOCK68 UNION ALL
SELECT * FROM STOCK69 UNION ALL
SELECT * FROM STOCK70 UNION ALL
SELECT * FROM STOCK71 UNION ALL
SELECT * FROM STOCK72 UNION ALL
SELECT * FROM STOCK73 UNION ALL
SELECT * FROM STOCK74 UNION ALL
SELECT * FROM STOCK75 UNION ALL
SELECT * FROM STOCK76 UNION ALL
SELECT * FROM STOCK77 UNION ALL
SELECT * FROM STOCK78 UNION ALL
SELECT * FROM STOCK79 UNION ALL
SELECT * FROM STOCK80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_WAREHOUSE.ddl

```
connect to TPCC in share mode;
DROP VIEW WAREHOUSE;
CREATE VIEW WAREHOUSE
            (W_NAME,
             W_STREET_1,
             W_STREET_2,
             W_CITY,
             W_STATE,
             W_ZIP,
             W_TAX,
             W_YTD,
             W_ID
            ) AS SELECT * FROM WAREHOUSE1 UNION ALL
SELECT * FROM WAREHOUSE2 UNION ALL
SELECT * FROM WAREHOUSE3 UNION ALL
SELECT * FROM WAREHOUSE4 UNION ALL
SELECT * FROM WAREHOUSE5 UNION ALL
SELECT * FROM WAREHOUSE6 UNION ALL
SELECT * FROM WAREHOUSE7 UNION ALL
SELECT * FROM WAREHOUSE8 UNION ALL
SELECT * FROM WAREHOUSE9 UNION ALL
SELECT * FROM WAREHOUSE10 UNION ALL
SELECT * FROM WAREHOUSE11 UNION ALL
SELECT * FROM WAREHOUSE12 UNION ALL
SELECT * FROM WAREHOUSE13 UNION ALL
SELECT * FROM WAREHOUSE14 UNION ALL
SELECT * FROM WAREHOUSE15 UNION ALL
SELECT * FROM WAREHOUSE16 UNION ALL
SELECT * FROM WAREHOUSE17 UNION ALL
SELECT * FROM WAREHOUSE18 UNION ALL
SELECT * FROM WAREHOUSE19 UNION ALL
SELECT * FROM WAREHOUSE20 UNION ALL
SELECT * FROM WAREHOUSE21 UNION ALL
SELECT * FROM WAREHOUSE22 UNION ALL
SELECT * FROM WAREHOUSE23 UNION ALL
SELECT * FROM WAREHOUSE24 UNION ALL
SELECT * FROM WAREHOUSE25 UNION ALL
SELECT * FROM WAREHOUSE26 UNION ALL
SELECT * FROM WAREHOUSE27 UNION ALL
SELECT * FROM WAREHOUSE28 UNION ALL
SELECT * FROM WAREHOUSE29 UNION ALL
SELECT * FROM WAREHOUSE30 UNION ALL
SELECT * FROM WAREHOUSE31 UNION ALL
```

```
SELECT * FROM WAREHOUSE32 UNION ALL
SELECT * FROM WAREHOUSE33 UNION ALL
SELECT * FROM WAREHOUSE34 UNION ALL
SELECT * FROM WAREHOUSE35 UNION ALL
SELECT * FROM WAREHOUSE36 UNION ALL
SELECT * FROM WAREHOUSE37 UNION ALL
SELECT * FROM WAREHOUSE38 UNION ALL
SELECT * FROM WAREHOUSE39 UNION ALL
SELECT * FROM WAREHOUSE40 UNION ALL
SELECT * FROM WAREHOUSE41 UNION ALL
SELECT * FROM WAREHOUSE42 UNION ALL
SELECT * FROM WAREHOUSE43 UNION ALL
SELECT * FROM WAREHOUSE44 UNION ALL
SELECT * FROM WAREHOUSE45 UNION ALL
SELECT * FROM WAREHOUSE46 UNION ALL
SELECT * FROM WAREHOUSE47 UNION ALL
SELECT * FROM WAREHOUSE48 UNION ALL
SELECT * FROM WAREHOUSE49 UNION ALL
SELECT * FROM WAREHOUSE50 UNION ALL
SELECT * FROM WAREHOUSE51 UNION ALL
SELECT * FROM WAREHOUSE52 UNION ALL
SELECT * FROM WAREHOUSE53 UNION ALL
SELECT * FROM WAREHOUSE54 UNION ALL
SELECT * FROM WAREHOUSE55 UNION ALL
SELECT * FROM WAREHOUSE56 UNION ALL
SELECT * FROM WAREHOUSE57 UNION ALL
SELECT * FROM WAREHOUSE58 UNION ALL
SELECT * FROM WAREHOUSE59 UNION ALL
SELECT * FROM WAREHOUSE60 UNION ALL
SELECT * FROM WAREHOUSE61 UNION ALL
SELECT * FROM WAREHOUSE62 UNION ALL
SELECT * FROM WAREHOUSE63 UNION ALL
SELECT * FROM WAREHOUSE64 UNION ALL
SELECT * FROM WAREHOUSE65 UNION ALL
SELECT * FROM WAREHOUSE66 UNION ALL
SELECT * FROM WAREHOUSE67 UNION ALL
SELECT * FROM WAREHOUSE68 UNION ALL
SELECT * FROM WAREHOUSE69 UNION ALL
SELECT * FROM WAREHOUSE70 UNION ALL
SELECT * FROM WAREHOUSE71 UNION ALL
SELECT * FROM WAREHOUSE72 UNION ALL
SELECT * FROM WAREHOUSE73 UNION ALL
SELECT * FROM WAREHOUSE74 UNION ALL
SELECT * FROM WAREHOUSE75 UNION ALL
SELECT * FROM WAREHOUSE76 UNION ALL
SELECT * FROM WAREHOUSE77 UNION ALL
SELECT * FROM WAREHOUSE78 UNION ALL
SELECT * FROM WAREHOUSE79 UNION ALL
SELECT * FROM WAREHOUSE80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/GEN_CUSTOMER_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 520 -f1 /flx8/flat_001/customer_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 521 1040 -f1 /flx8/flat_002/customer_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1041 1560 -f1
/flx8/flat_003/customer_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1561 2080 -f1
/flx8/flat_004/customer_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2081 2600 -f1
/flx8/flat_005/customer_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2601 3120 -f1
/flx8/flat_006/customer_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3121 3640 -f1
/flx8/flat_007/customer_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3641 4160 -f1
/flx8/flat_008/customer_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4161 4680 -f1
/flx8/flat_009/customer_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4681 5200 -f1
/flx8/flat_010/customer_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5201 5720 -f1
/flx8/flat_011/customer_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5721 6240 -f1
/flx8/flat_012/customer_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6241 6760 -f1
/flx8/flat_013/customer_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6761 7280 -f1
/flx8/flat_014/customer_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7281 7800 -f1
/flx8/flat_015/customer_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7801 8320 -f1
/flx8/flat_016/customer_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8321 8840 -f1
/flx8/flat_017/customer_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8841 9360 -f1
/flx8/flat_018/customer_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9361 9880 -f1
/flx8/flat_019/customer_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9881 10400 -f1
/flx8/flat_020/customer_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10401 10920 -f1
/flx8/flat_021/customer_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10921 11440 -f1
/flx8/flat_022/customer_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11441 11960 -f1
/flx8/flat_023/customer_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11961 12480 -f1
/flx8/flat_024/customer_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12481 13000 -f1
/flx8/flat_025/customer_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13001 13520 -f1
/flx8/flat_026/customer_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13521 14040 -f1
/flx8/flat_027/customer_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14041 14560 -f1
/flx8/flat_028/customer_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14561 15080 -f1
/flx8/flat_029/customer_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15081 15600 -f1
/flx8/flat_030/customer_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15601 16120 -f1
/flx8/flat_031/customer_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16121 16640 -f1
/flx8/flat_032/customer_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16641 17160 -f1
/flx8/flat_033/customer_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17161 17680 -f1
/flx8/flat_034/customer_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17681 18200 -f1
/flx8/flat_035/customer_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18201 18720 -f1
/flx8/flat_036/customer_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18721 19240 -f1
/flx8/flat_037/customer_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19241 19760 -f1
/flx8/flat_038/customer_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19761 20280 -f1
/flx8/flat_039/customer_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20281 20800 -f1
/flx8/flat_040/customer_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20801 21320 -f1
/flx8/flat_041/customer_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21321 21840 -f1
/flx8/flat_042/customer_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21841 22360 -f1
/flx8/flat_043/customer_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22361 22880 -f1
/flx8/flat_044/customer_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22881 23400 -f1
/flx8/flat_045/customer_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23401 23920 -f1
/flx8/flat_046/customer_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23921 24440 -f1
/flx8/flat_047/customer_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24441 24960 -f1
/flx8/flat_048/customer_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24961 25480 -f1
/flx8/flat_049/customer_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25481 26000 -f1
/flx8/flat_050/customer_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26001 26520 -f1
/flx8/flat_051/customer_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26521 27040 -f1
/flx8/flat_052/customer_052_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27041 27560 -f1
/flx8/flat_053/customer_053_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27561 28080 -f1
/flx8/flat_054/customer_054_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28081 28600 -f1
/flx8/flat_055/customer_055_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28601 29120 -f1
/flx8/flat_056/customer_056_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 29121 29640 -f1
/flx8/flat_057/customer_057_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 29641 30160 -f1
/flx8/flat_058/customer_058_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30161 30680 -f1
/flx8/flat_059/customer_059_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30681 31200 -f1
/flx8/flat_060/customer_060_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31201 31720 -f1
/flx8/flat_061/customer_061_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31721 32240 -f1
/flx8/flat_062/customer_062_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32241 32760 -f1
/flx8/flat_063/customer_063_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32761 33280 -f1
/flx8/flat_064/customer_064_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33281 33800 -f1
/flx8/flat_065/customer_065_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33801 34320 -f1
/flx8/flat_066/customer_066_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34321 34840 -f1
/flx8/flat_067/customer_067_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34841 35360 -f1
/flx8/flat_068/customer_068_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35361 35880 -f1
/flx8/flat_069/customer_069_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35881 36400 -f1
/flx8/flat_070/customer_070_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36401 36920 -f1
/flx8/flat_071/customer_071_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36921 37440 -f1
/flx8/flat_072/customer_072_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37441 37960 -f1
/flx8/flat_073/customer_073_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37961 38480 -f1
/flx8/flat_074/customer_074_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38481 39000 -f1
/flx8/flat_075/customer_075_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39001 39520 -f1
/flx8/flat_076/customer_076_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39521 40040 -f1
/flx8/flat_077/customer_077_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40041 40560 -f1
/flx8/flat_078/customer_078_1.dat
```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40561 41080 -f1
/flx8/flat_079/customer_079_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 41081 41600 -f1
/flx8/flat_080/customer_080_1.dat

DDL/GEN_DISTRICT_ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1 520 -f1 /flx8/flat_001/district_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 521 1040 -f1 /flx8/flat_002/district_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1041 1560 -f1 /flx8/flat_003/district_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1561 2080 -f1 /flx8/flat_004/district_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 2081 2600 -f1 /flx8/flat_005/district_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 2601 3120 -f1 /flx8/flat_006/district_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 3121 3640 -f1 /flx8/flat_007/district_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 3641 4160 -f1 /flx8/flat_008/district_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 4161 4680 -f1 /flx8/flat_009/district_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 4681 5200 -f1 /flx8/flat_010/district_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 5201 5720 -f1 /flx8/flat_011/district_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 5721 6240 -f1 /flx8/flat_012/district_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 6241 6760 -f1 /flx8/flat_013/district_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 6761 7280 -f1 /flx8/flat_014/district_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 7281 7800 -f1 /flx8/flat_015/district_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 7801 8320 -f1 /flx8/flat_016/district_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 8321 8840 -f1 /flx8/flat_017/district_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 8841 9360 -f1 /flx8/flat_018/district_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 9361 9880 -f1 /flx8/flat_019/district_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 9881 10400 -f1 /flx8/flat_020/district_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 10401 10920 -f1
/flx8/flat_021/district_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 10921 11440 -f1
/flx8/flat_022/district_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 11441 11960 -f1
/flx8/flat_023/district_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 11961 12480 -f1
/flx8/flat_024/district_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 12481 13000 -f1
/flx8/flat_025/district_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 13001 13520 -f1
/flx8/flat_026/district_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 13521 14040 -f1
/flx8/flat_027/district_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 14041 14560 -f1
/flx8/flat_028/district_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 14561 15080 -f1
/flx8/flat_029/district_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 15081 15600 -f1
/flx8/flat_030/district_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 15601 16120 -f1
/flx8/flat_031/district_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 16121 16640 -f1
/flx8/flat_032/district_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 16641 17160 -f1
/flx8/flat_033/district_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 17161 17680 -f1
/flx8/flat_034/district_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 17681 18200 -f1
/flx8/flat_035/district_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 18201 18720 -f1
/flx8/flat_036/district_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 18721 19240 -f1
/flx8/flat_037/district_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 19241 19760 -f1
/flx8/flat_038/district_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 19761 20280 -f1
/flx8/flat_039/district_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 20281 20800 -f1
/flx8/flat_040/district_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 20801 21320 -f1
/flx8/flat_041/district_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 21321 21840 -f1
/flx8/flat_042/district_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 21841 22360 -f1
/flx8/flat_043/district_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 22361 22880 -f1
/flx8/flat_044/district_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 22881 23400 -f1
/flx8/flat_045/district_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 23401 23920 -f1
/flx8/flat_046/district_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 23921 24440 -f1
/flx8/flat_047/district_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 24441 24960 -f1
/flx8/flat_048/district_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 24961 25480 -f1
/flx8/flat_049/district_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 25481 26000 -f1
/flx8/flat_050/district_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 26001 26520 -f1
/flx8/flat_051/district_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 26521 27040 -f1
/flx8/flat_052/district_052_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 27041 27560 -f1
/flx8/flat_053/district_053_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 27561 28080 -f1
/flx8/flat_054/district_054_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 28081 28600 -f1
/flx8/flat_055/district_055_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 28601 29120 -f1
/flx8/flat_056/district_056_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 29121 29640 -f1
/flx8/flat_057/district_057_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 29641 30160 -f1
/flx8/flat_058/district_058_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 30161 30680 -f1
/flx8/flat_059/district_059_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 30681 31200 -f1
/flx8/flat_060/district_060_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 31201 31720 -f1
/flx8/flat_061/district_061_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 31721 32240 -f1
/flx8/flat_062/district_062_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 32241 32760 -f1
/flx8/flat_063/district_063_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 32761 33280 -f1
/flx8/flat_064/district_064_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 33281 33800 -f1
/flx8/flat_065/district_065_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 33801 34320 -f1
/flx8/flat_066/district_066_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 34321 34840 -f1
/flx8/flat_067/district_067_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 34841 35360 -f1
/flx8/flat_068/district_068_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 35361 35880 -f1
/flx8/flat_069/district_069_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 35881 36400 -f1
/flx8/flat_070/district_070_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 36401 36920 -f1
/flx8/flat_071/district_071_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 36921 37440 -f1
/flx8/flat_072/district_072_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 37441 37960 -f1
/flx8/flat_073/district_073_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 37961 38480 -f1
/flx8/flat_074/district_074_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 38481 39000 -f1
/flx8/flat_075/district_075_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 39001 39520 -f1
/flx8/flat_076/district_076_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 39521 40040 -f1
/flx8/flat_077/district_077_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 40041 40560 -f1
/flx8/flat_078/district_078_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 40561 41080 -f1
/flx8/flat_079/district_079_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 41081 41600 -f1
/flx8/flat_080/district_080_1.dat

DDL/GEN_HISTORY_ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1 520 -f1 /flx8/flat_001/history_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 521 1040 -f1 /flx8/flat_002/history_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1041 1560 -f1 /flx8/flat_003/history_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1561 2080 -f1 /flx8/flat_004/history_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 2081 2600 -f1 /flx8/flat_005/history_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 2601 3120 -f1 /flx8/flat_006/history_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 3121 3640 -f1 /flx8/flat_007/history_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 3641 4160 -f1 /flx8/flat_008/history_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 4161 4680 -f1 /flx8/flat_009/history_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 4681 5200 -f1 /flx8/flat_010/history_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 5201 5720 -f1 /flx8/flat_011/history_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 5721 6240 -f1 /flx8/flat_012/history_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 6241 6760 -f1 /flx8/flat_013/history_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 6761 7280 -f1 /flx8/flat_014/history_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 7281 7800 -f1 /flx8/flat_015/history_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 7801 8320 -f1 /flx8/flat_016/history_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 8321 8840 -f1 /flx8/flat_017/history_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 8841 9360 -f1 /flx8/flat_018/history_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 9361 9880 -f1 /flx8/flat_019/history_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 9881 10400 -f1 /flx8/flat_020/history_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 10401 10920 -f1
/flx8/flat_021/history_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 10921 11440 -f1
/flx8/flat_022/history_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 11441 11960 -f1
/flx8/flat_023/history_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 11961 12480 -f1
/flx8/flat_024/history_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 12481 13000 -f1
/flx8/flat_025/history_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 13001 13520 -f1
/flx8/flat_026/history_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 13521 14040 -f1
/flx8/flat_027/history_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 14041 14560 -f1
/flx8/flat_028/history_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 14561 15080 -f1
/flx8/flat_029/history_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 15081 15600 -f1
/flx8/flat_030/history_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 15601 16120 -f1
/flx8/flat_031/history_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 16121 16640 -f1
/flx8/flat_032/history_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 16641 17160 -f1
/flx8/flat_033/history_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 17161 17680 -f1
/flx8/flat_034/history_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 17681 18200 -f1
/flx8/flat_035/history_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 18201 18720 -f1
/flx8/flat_036/history_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 18721 19240 -f1
/flx8/flat_037/history_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 19241 19760 -f1
/flx8/flat_038/history_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 19761 20280 -f1
/flx8/flat_039/history_039_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 20281 20800 -f1 /flx8/flat_040/history_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 20801 21320 -f1 /flx8/flat_041/history_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 21321 21840 -f1 /flx8/flat_042/history_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 21841 22360 -f1 /flx8/flat_043/history_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 22361 22880 -f1 /flx8/flat_044/history_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 22881 23400 -f1 /flx8/flat_045/history_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 23401 23920 -f1 /flx8/flat_046/history_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 23921 24440 -f1 /flx8/flat_047/history_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 24441 24960 -f1 /flx8/flat_048/history_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 24961 25480 -f1 /flx8/flat_049/history_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 25481 26000 -f1 /flx8/flat_050/history_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 26001 26520 -f1 /flx8/flat_051/history_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 26521 27040 -f1 /flx8/flat_052/history_052_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 27041 27560 -f1 /flx8/flat_053/history_053_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 27561 28080 -f1 /flx8/flat_054/history_054_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 28081 28600 -f1 /flx8/flat_055/history_055_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 28601 29120 -f1 /flx8/flat_056/history_056_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 29121 29640 -f1 /flx8/flat_057/history_057_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 29641 30160 -f1 /flx8/flat_058/history_058_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 30161 30680 -f1 /flx8/flat_059/history_059_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 30681 31200 -f1 /flx8/flat_060/history_060_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 31201 31720 -f1 /flx8/flat_061/history_061_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 31721 32240 -f1 /flx8/flat_062/history_062_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 32241 32760 -f1 /flx8/flat_063/history_063_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 32761 33280 -f1 /flx8/flat_064/history_064_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 33281 33800 -f1 /flx8/flat_065/history_065_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 33801 34320 -f1 /flx8/flat_066/history_066_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 34321 34840 -f1 /flx8/flat_067/history_067_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 34841 35360 -f1 /flx8/flat_068/history_068_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 35361 35880 -f1 /flx8/flat_069/history_069_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 35881 36400 -f1 /flx8/flat_070/history_070_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 36401 36920 -f1 /flx8/flat_071/history_071_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 36921 37440 -f1 /flx8/flat_072/history_072_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 37441 37960 -f1 /flx8/flat_073/history_073_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 37961 38480 -f1 /flx8/flat_074/history_074_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 38481 39000 -f1 /flx8/flat_075/history_075_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 39001 39520 -f1 /flx8/flat_076/history_076_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 39521 40040 -f1 /flx8/flat_077/history_077_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 40041 40560 -f1 /flx8/flat_078/history_078_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 40561 41080 -f1 /flx8/flat_079/history_079_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 41081 41600 -f1 /flx8/flat_080/history_080_1.dat

DDL/GEN_ITEM_ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 5 -f1 /flx8/flat/item_1.dat

DDL/GEN_NEW_ORDER_ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1 520 -f1 /flx8/flat_001/neworder_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 521 1040 -f1 /flx8/flat_002/neworder_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1041 1560 -f1 /flx8/flat_003/neworder_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1561 2080 -f1 /flx8/flat_004/neworder_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 2081 2600 -f1 /flx8/flat_005/neworder_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 2601 3120 -f1 /flx8/flat_006/neworder_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 3121 3640 -f1 /flx8/flat_007/neworder_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 3641 4160 -f1 /flx8/flat_008/neworder_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 4161 4680 -f1 /flx8/flat_009/neworder_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 4681 5200 -f1 /flx8/flat_010/neworder_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 5201 5720 -f1 /flx8/flat_011/neworder_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 5721 6240 -f1 /flx8/flat_012/neworder_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 6241 6760 -f1 /flx8/flat_013/neworder_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 6761 7280 -f1 /flx8/flat_014/neworder_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 7281 7800 -f1 /flx8/flat_015/neworder_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 7801 8320 -f1 /flx8/flat_016/neworder_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 8321 8840 -f1 /flx8/flat_017/neworder_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 8841 9360 -f1 /flx8/flat_018/neworder_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 9361 9880 -f1 /flx8/flat_019/neworder_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 9881 10400 -f1 /flx8/flat_020/neworder_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 10401 10920 -f1 /flx8/flat_021/neworder_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 10921 11440 -f1 /flx8/flat_022/neworder_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 11441 11960 -f1 /flx8/flat_023/neworder_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 11961 12480 -f1 /flx8/flat_024/neworder_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 12481 13000 -f1 /flx8/flat_025/neworder_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 13001 13520 -f1 /flx8/flat_026/neworder_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 13521 14040 -f1 /flx8/flat_027/neworder_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 14041 14560 -f1 /flx8/flat_028/neworder_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 14561 15080 -f1 /flx8/flat_029/neworder_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 15081 15600 -f1 /flx8/flat_030/neworder_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 15601 16120 -f1 /flx8/flat_031/neworder_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 16121 16640 -f1 /flx8/flat_032/neworder_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 16641 17160 -f1 /flx8/flat_033/neworder_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 17161 17680 -f1 /flx8/flat_034/neworder_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 17681 18200 -f1 /flx8/flat_035/neworder_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 18201 18720 -f1 /flx8/flat_036/neworder_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 18721 19240 -f1 /flx8/flat_037/neworder_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 19241 19760 -f1 /flx8/flat_038/neworder_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 19761 20280 -f1 /flx8/flat_039/neworder_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 20281 20800 -f1 /flx8/flat_040/neworder_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 20801 21320 -f1 /flx8/flat_041/neworder_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 21321 21840 -f1 /flx8/flat_042/neworder_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 21841 22360 -f1 /flx8/flat_043/neworder_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 22361 22880 -f1 /flx8/flat_044/neworder_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 22881 23400 -f1 /flx8/flat_045/neworder_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 23401 23920 -f1 /flx8/flat_046/neworder_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 23921 24440 -f1 /flx8/flat_047/neworder_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 24441 24960 -f1 /flx8/flat_048/neworder_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 24961 25480 -f1 /flx8/flat_049/neworder_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 25481 26000 -f1 /flx8/flat_050/neworder_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 26001 26520 -f1 /flx8/flat_051/neworder_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 26521 27040 -f1 /flx8/flat_052/neworder_052_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 27041 27560 -f1 /flx8/flat_053/neworder_053_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 27561 28080 -f1 /flx8/flat_054/neworder_054_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 28081 28600 -f1 /flx8/flat_055/neworder_055_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 28601 29120 -f1 /flx8/flat_056/neworder_056_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 29121 29640 -f1 /flx8/flat_057/neworder_057_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 29641 30160 -f1 /flx8/flat_058/neworder_058_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 30161 30680 -f1 /flx8/flat_059/neworder_059_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 30681 31200 -f1 /flx8/flat_060/neworder_060_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 31201 31720 -f1 /flx8/flat_061/neworder_061_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 31721 32240 -f1 /flx8/flat_062/neworder_062_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 32241 32760 -f1 /flx8/flat_063/neworder_063_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 32761 33280 -f1 /flx8/flat_064/neworder_064_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 33281 33800 -f1 /flx8/flat_065/neworder_065_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 33801 34320 -f1 /flx8/flat_066/neworder_066_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 34321 34840 -f1 /flx8/flat_067/neworder_067_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 34841 35360 -f1 /flx8/flat_068/neworder_068_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 35361 35880 -f1 /flx8/flat_069/neworder_069_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 35881 36400 -f1 /flx8/flat_070/neworder_070_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 36401 36920 -f1 /flx8/flat_071/neworder_071_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 36921 37440 -f1 /flx8/flat_072/neworder_072_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 37441 37960 -f1 /flx8/flat_073/neworder_073_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 37961 38480 -f1 /flx8/flat_074/neworder_074_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 38481 39000 -f1 /flx8/flat_075/neworder_075_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 39001 39520 -f1 /flx8/flat_076/neworder_076_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 39521 40040 -f1 /flx8/flat_077/neworder_077_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 40041 40560 -f1 /flx8/flat_078/neworder_078_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 40561 41080 -f1 /flx8/flat_079/neworder_079_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 41081 41600 -f1 /flx8/flat_080/neworder_080_1.dat

DDL/GEN_ORDERS_ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1 520 -f1 /flx8/flat_001/orders_001_1.dat -f2 /flx8/flat_001/orderline_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 521 1040 -f1 /flx8/flat_002/orders_002_1.dat -f2 /flx8/flat_002/orderline_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1041 1560 -f1 /flx8/flat_003/orders_003_1.dat -f2 /flx8/flat_003/orderline_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1561 2080 -f1 /flx8/flat_004/orders_004_1.dat -f2 /flx8/flat_004/orderline_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2081 2600 -f1 /flx8/flat_005/orders_005_1.dat -f2 /flx8/flat_005/orderline_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2601 3120 -f1 /flx8/flat_006/orders_006_1.dat -f2 /flx8/flat_006/orderline_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3121 3640 -f1 /flx8/flat_007/orders_007_1.dat -f2 /flx8/flat_007/orderline_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3641 4160 -f1 /flx8/flat_008/orders_008_1.dat -f2 /flx8/flat_008/orderline_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4161 4680 -f1 /flx8/flat_009/orders_009_1.dat -f2 /flx8/flat_009/orderline_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4681 5200 -f1 /flx8/flat_010/orders_010_1.dat -f2 /flx8/flat_010/orderline_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5201 5720 -f1 /flx8/flat_011/orders_011_1.dat -f2 /flx8/flat_011/orderline_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5721 6240 -f1 /flx8/flat_012/orders_012_1.dat -f2 /flx8/flat_012/orderline_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6241 6760 -f1 /flx8/flat_013/orders_013_1.dat -f2 /flx8/flat_013/orderline_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6761 7280 -f1 /flx8/flat_014/orders_014_1.dat -f2 /flx8/flat_014/orderline_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7281 7800 -f1 /flx8/flat_015/orders_015_1.dat -f2 /flx8/flat_015/orderline_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7801 8320 -f1 /flx8/flat_016/orders_016_1.dat -f2 /flx8/flat_016/orderline_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8321 8840 -f1 /flx8/flat_017/orders_017_1.dat -f2 /flx8/flat_017/orderline_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8841 9360 -f1 /flx8/flat_018/orders_018_1.dat -f2 /flx8/flat_018/orderline_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9361 9880 -f1 /flx8/flat_019/orders_019_1.dat -f2 /flx8/flat_019/orderline_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9881 10400 -f1 /flx8/flat_020/orders_020_1.dat -f2 /flx8/flat_020/orderline_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10401 10920 -f1 /flx8/flat_021/orders_021_1.dat -f2 /flx8/flat_021/orderline_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10921 11440 -f1 /flx8/flat_022/orders_022_1.dat -f2 /flx8/flat_022/orderline_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11441 11960 -f1 /flx8/flat_023/orders_023_1.dat -f2 /flx8/flat_023/orderline_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11961 12480 -f1 /flx8/flat_024/orders_024_1.dat -f2 /flx8/flat_024/orderline_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 12481 13000 -f1 /flx8/flat_025/orders_025_1.dat -f2 /flx8/flat_025/orderline_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13001 13520 -f1 /flx8/flat_026/orders_026_1.dat -f2 /flx8/flat_026/orderline_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13521 14040 -f1 /flx8/flat_027/orders_027_1.dat -f2 /flx8/flat_027/orderline_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14041 14560 -f1 /flx8/flat_028/orders_028_1.dat -f2 /flx8/flat_028/orderline_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14561 15080 -f1 /flx8/flat_029/orders_029_1.dat -f2 /flx8/flat_029/orderline_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15081 15600 -f1 /flx8/flat_030/orders_030_1.dat -f2 /flx8/flat_030/orderline_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15601 16120 -f1 /flx8/flat_031/orders_031_1.dat -f2 /flx8/flat_031/orderline_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16121 16640 -f1 /flx8/flat_032/orders_032_1.dat -f2 /flx8/flat_032/orderline_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16641 17160 -f1 /flx8/flat_033/orders_033_1.dat -f2 /flx8/flat_033/orderline_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17161 17680 -f1 /flx8/flat_034/orders_034_1.dat -f2 /flx8/flat_034/orderline_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17681 18200 -f1 /flx8/flat_035/orders_035_1.dat -f2 /flx8/flat_035/orderline_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18201 18720 -f1 /flx8/flat_036/orders_036_1.dat -f2 /flx8/flat_036/orderline_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18721 19240 -f1 /flx8/flat_037/orders_037_1.dat -f2 /flx8/flat_037/orderline_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19241 19760 -f1 /flx8/flat_038/orders_038_1.dat -f2 /flx8/flat_038/orderline_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19761 20280 -f1 /flx8/flat_039/orders_039_1.dat -f2 /flx8/flat_039/orderline_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20281 20800 -f1 /flx8/flat_040/orders_040_1.dat -f2 /flx8/flat_040/orderline_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20801 21320 -f1 /flx8/flat_041/orders_041_1.dat -f2 /flx8/flat_041/orderline_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21321 21840 -f1 /flx8/flat_042/orders_042_1.dat -f2 /flx8/flat_042/orderline_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21841 22360 -f1 /flx8/flat_043/orders_043_1.dat -f2 /flx8/flat_043/orderline_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22361 22880 -f1 /flx8/flat_044/orders_044_1.dat -f2 /flx8/flat_044/orderline_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22881 23400 -f1 /flx8/flat_045/orders_045_1.dat -f2 /flx8/flat_045/orderline_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23401 23920 -f1 /flx8/flat_046/orders_046_1.dat -f2 /flx8/flat_046/orderline_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23921 24440 -f1 /flx8/flat_047/orders_047_1.dat -f2 /flx8/flat_047/orderline_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24441 24960 -f1 /flx8/flat_048/orders_048_1.dat -f2 /flx8/flat_048/orderline_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24961 25480 -f1 /flx8/flat_049/orders_049_1.dat -f2 /flx8/flat_049/orderline_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 25481 26000 -f1 /flx8/flat_050/orders_050_1.dat -f2 /flx8/flat_050/orderline_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26001 26520 -f1 /flx8/flat_051/orders_051_1.dat -f2 /flx8/flat_051/orderline_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26521 27040 -f1 /flx8/flat_052/orders_052_1.dat -f2 /flx8/flat_052/orderline_052_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27041 27560 -f1 /flx8/flat_053/orders_053_1.dat -f2 /flx8/flat_053/orderline_053_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27561 28080 -f1 /flx8/flat_054/orders_054_1.dat -f2 /flx8/flat_054/orderline_054_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28081 28600 -f1 /flx8/flat_055/orders_055_1.dat -f2 /flx8/flat_055/orderline_055_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28601 29120 -f1 /flx8/flat_056/orders_056_1.dat -f2 /flx8/flat_056/orderline_056_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 29121 29640 -f1 /flx8/flat_057/orders_057_1.dat -f2 /flx8/flat_057/orderline_057_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 29641 30160 -f1 /flx8/flat_058/orders_058_1.dat -f2 /flx8/flat_058/orderline_058_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30161 30680 -f1 /flx8/flat_059/orders_059_1.dat -f2 /flx8/flat_059/orderline_059_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30681 31200 -f1 /flx8/flat_060/orders_060_1.dat -f2 /flx8/flat_060/orderline_060_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31201 31720 -f1 /flx8/flat_061/orders_061_1.dat -f2 /flx8/flat_061/orderline_061_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31721 32240 -f1 /flx8/flat_062/orders_062_1.dat -f2 /flx8/flat_062/orderline_062_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32241 32760 -f1 /flx8/flat_063/orders_063_1.dat -f2 /flx8/flat_063/orderline_063_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32761 33280 -f1 /flx8/flat_064/orders_064_1.dat -f2 /flx8/flat_064/orderline_064_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33281 33800 -f1 /flx8/flat_065/orders_065_1.dat -f2 /flx8/flat_065/orderline_065_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33801 34320 -f1 /flx8/flat_066/orders_066_1.dat -f2 /flx8/flat_066/orderline_066_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34321 34840 -f1 /flx8/flat_067/orders_067_1.dat -f2 /flx8/flat_067/orderline_067_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34841 35360 -f1 /flx8/flat_068/orders_068_1.dat -f2 /flx8/flat_068/orderline_068_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35361 35880 -f1 /flx8/flat_069/orders_069_1.dat -f2 /flx8/flat_069/orderline_069_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35881 36400 -f1 /flx8/flat_070/orders_070_1.dat -f2 /flx8/flat_070/orderline_070_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36401 36920 -f1 /flx8/flat_071/orders_071_1.dat -f2 /flx8/flat_071/orderline_071_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36921 37440 -f1 /flx8/flat_072/orders_072_1.dat -f2 /flx8/flat_072/orderline_072_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37441 37960 -f1 /flx8/flat_073/orders_073_1.dat -f2 /flx8/flat_073/orderline_073_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37961 38480 -f1 /flx8/flat_074/orders_074_1.dat -f2 /flx8/flat_074/orderline_074_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38481 39000 -f1 /flx8/flat_075/orders_075_1.dat -f2 /flx8/flat_075/orderline_075_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39001 39520 -f1 /flx8/flat_076/orders_076_1.dat -f2 /flx8/flat_076/orderline_076_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39521 40040 -f1 /flx8/flat_077/orders_077_1.dat -f2 /flx8/flat_077/orderline_077_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40041 40560 -f1 /flx8/flat_078/orders_078_1.dat -f2 /flx8/flat_078/orderline_078_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40561 41080 -f1 /flx8/flat_079/orders_079_1.dat -f2 /flx8/flat_079/orderline_079_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 41081 41600 -f1 /flx8/flat_080/orders_080_1.dat -f2 /flx8/flat_080/orderline_080_1.dat

DDL/GEN_STOCK_ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1 520 -f1 /flx8/flat_001/stock_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 521 1040 -f1 /flx8/flat_002/stock_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1041 1560 -f1 /flx8/flat_003/stock_003_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1561 2080 -f1 /flx8/flat_004/stock_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2081 2600 -f1 /flx8/flat_005/stock_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2601 3120 -f1 /flx8/flat_006/stock_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3121 3640 -f1 /flx8/flat_007/stock_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3641 4160 -f1 /flx8/flat_008/stock_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4161 4680 -f1 /flx8/flat_009/stock_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4681 5200 -f1 /flx8/flat_010/stock_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5201 5720 -f1 /flx8/flat_011/stock_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5721 6240 -f1 /flx8/flat_012/stock_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6241 6760 -f1 /flx8/flat_013/stock_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6761 7280 -f1 /flx8/flat_014/stock_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7281 7800 -f1 /flx8/flat_015/stock_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7801 8320 -f1 /flx8/flat_016/stock_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8321 8840 -f1 /flx8/flat_017/stock_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8841 9360 -f1 /flx8/flat_018/stock_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9361 9880 -f1 /flx8/flat_019/stock_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9881 10400 -f1 /flx8/flat_020/stock_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10401 10920 -f1 /flx8/flat_021/stock_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10921 11440 -f1 /flx8/flat_022/stock_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11441 11960 -f1 /flx8/flat_023/stock_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11961 12480 -f1 /flx8/flat_024/stock_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12481 13000 -f1 /flx8/flat_025/stock_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13001 13520 -f1 /flx8/flat_026/stock_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13521 14040 -f1 /flx8/flat_027/stock_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14041 14560 -f1 /flx8/flat_028/stock_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14561 15080 -f1 /flx8/flat_029/stock_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15081 15600 -f1 /flx8/flat_030/stock_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15601 16120 -f1 /flx8/flat_031/stock_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16121 16640 -f1 /flx8/flat_032/stock_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16641 17160 -f1 /flx8/flat_033/stock_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 17161 17680 -f1 /flx8/flat_034/stock_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 17681 18200 -f1 /flx8/flat_035/stock_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18201 18720 -f1 /flx8/flat_036/stock_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18721 19240 -f1 /flx8/flat_037/stock_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19241 19760 -f1 /flx8/flat_038/stock_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19761 20280 -f1 /flx8/flat_039/stock_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20281 20800 -f1 /flx8/flat_040/stock_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20801 21320 -f1 /flx8/flat_041/stock_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21321 21840 -f1 /flx8/flat_042/stock_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21841 22360 -f1 /flx8/flat_043/stock_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22361 22880 -f1 /flx8/flat_044/stock_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22881 23400 -f1 /flx8/flat_045/stock_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23401 23920 -f1 /flx8/flat_046/stock_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23921 24440 -f1 /flx8/flat_047/stock_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24441 24960 -f1 /flx8/flat_048/stock_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24961 25480 -f1 /flx8/flat_049/stock_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 25481 26000 -f1 /flx8/flat_050/stock_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26001 26520 -f1 /flx8/flat_051/stock_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26521 27040 -f1 /flx8/flat_052/stock_052_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27041 27560 -f1 /flx8/flat_053/stock_053_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27561 28080 -f1 /flx8/flat_054/stock_054_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28081 28600 -f1 /flx8/flat_055/stock_055_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28601 29120 -f1 /flx8/flat_056/stock_056_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 29121 29640 -f1 /flx8/flat_057/stock_057_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 29641 30160 -f1 /flx8/flat_058/stock_058_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30161 30680 -f1 /flx8/flat_059/stock_059_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30681 31200 -f1 /flx8/flat_060/stock_060_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31201 31720 -f1 /flx8/flat_061/stock_061_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31721 32240 -f1 /flx8/flat_062/stock_062_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32241 32760 -f1 /flx8/flat_063/stock_063_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32761 33280 -f1 /flx8/flat_064/stock_064_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33281 33800 -f1 /flx8/flat_065/stock_065_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33801 34320 -f1 /flx8/flat_066/stock_066_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34321 34840 -f1 /flx8/flat_067/stock_067_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34841 35360 -f1 /flx8/flat_068/stock_068_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 35361 35880 -f1 /flx8/flat_069/stock_069_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 35881 36400 -f1 /flx8/flat_070/stock_070_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36401 36920 -f1 /flx8/flat_071/stock_071_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36921 37440 -f1 /flx8/flat_072/stock_072_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 37441 37960 -f1 /flx8/flat_073/stock_073_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 37961 38480 -f1 /flx8/flat_074/stock_074_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38481 39000 -f1 /flx8/flat_075/stock_075_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39001 39520 -f1 /flx8/flat_076/stock_076_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39521 40040 -f1 /flx8/flat_077/stock_077_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40041 40560 -f1 /flx8/flat_078/stock_078_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40561 41080 -f1 /flx8/flat_079/stock_079_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 41081 41600 -f1 /flx8/flat_080/stock_080_1.dat

DDL/GEN_WAREHOUSE_ALL.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1 520 -f1 /flx8/flat_001/warehouse_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 521 1040 -f1
/flx8/flat_002/warehouse_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1041 1560 -f1
/flx8/flat_003/warehouse_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1561 2080 -f1
/flx8/flat_004/warehouse_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 2081 2600 -f1
/flx8/flat_005/warehouse_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 2601 3120 -f1
/flx8/flat_006/warehouse_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 3121 3640 -f1
/flx8/flat_007/warehouse_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 3641 4160 -f1
/flx8/flat_008/warehouse_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 4161 4680 -f1
/flx8/flat_009/warehouse_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 4681 5200 -f1
/flx8/flat_010/warehouse_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 5201 5720 -f1
/flx8/flat_011/warehouse_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 5721 6240 -f1
/flx8/flat_012/warehouse_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 6241 6760 -f1
/flx8/flat_013/warehouse_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 6761 7280 -f1
/flx8/flat_014/warehouse_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 7281 7800 -f1
/flx8/flat_015/warehouse_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 7801 8320 -f1
/flx8/flat_016/warehouse_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 8321 8840 -f1
/flx8/flat_017/warehouse_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 8841 9360 -f1
/flx8/flat_018/warehouse_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 9361 9880 -f1
/flx8/flat_019/warehouse_019_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 9881 10400 -f1
/flx8/flat_020/warehouse_020_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 10401 10920 -f1
/flx8/flat_021/warehouse_021_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 10921 11440 -f1
/flx8/flat_022/warehouse_022_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 11441 11960 -f1
/flx8/flat_023/warehouse_023_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 11961 12480 -f1
/flx8/flat_024/warehouse_024_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 12481 13000 -f1
/flx8/flat_025/warehouse_025_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 13001 13520 -f1
/flx8/flat_026/warehouse_026_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 13521 14040 -f1
/flx8/flat_027/warehouse_027_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 14041 14560 -f1
/flx8/flat_028/warehouse_028_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 14561 15080 -f1
/flx8/flat_029/warehouse_029_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 15081 15600 -f1
/flx8/flat_030/warehouse_030_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 15601 16120 -f1
/flx8/flat_031/warehouse_031_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 16121 16640 -f1
/flx8/flat_032/warehouse_032_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 16641 17160 -f1
/flx8/flat_033/warehouse_033_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 17161 17680 -f1
/flx8/flat_034/warehouse_034_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 17681 18200 -f1
/flx8/flat_035/warehouse_035_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 18201 18720 -f1
/flx8/flat_036/warehouse_036_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 18721 19240 -f1
/flx8/flat_037/warehouse_037_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 19241 19760 -f1
/flx8/flat_038/warehouse_038_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 19761 20280 -f1
/flx8/flat_039/warehouse_039_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 20281 20800 -f1
/flx8/flat_040/warehouse_040_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 20801 21320 -f1
/flx8/flat_041/warehouse_041_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 21321 21840 -f1
/flx8/flat_042/warehouse_042_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 21841 22360 -f1
/flx8/flat_043/warehouse_043_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 22361 22880 -f1
/flx8/flat_044/warehouse_044_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 22881 23400 -f1
/flx8/flat_045/warehouse_045_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 23401 23920 -f1
/flx8/flat_046/warehouse_046_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 23921 24440 -f1
/flx8/flat_047/warehouse_047_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 24441 24960 -f1
/flx8/flat_048/warehouse_048_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 24961 25480 -f1
/flx8/flat_049/warehouse_049_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 25481 26000 -f1
/flx8/flat_050/warehouse_050_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 26001 26520 -f1
/flx8/flat_051/warehouse_051_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 26521 27040 -f1
/flx8/flat_052/warehouse_052_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 27041 27560 -f1
/flx8/flat_053/warehouse_053_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 27561 28080 -f1
/flx8/flat_054/warehouse_054_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 28081 28600 -f1
/flx8/flat_055/warehouse_055_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 28601 29120 -f1
/flx8/flat_056/warehouse_056_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 29121 29640 -f1
/flx8/flat_057/warehouse_057_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 29641 30160 -f1
/flx8/flat_058/warehouse_058_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 30161 30680 -f1
/flx8/flat_059/warehouse_059_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 30681 31200 -f1
/flx8/flat_060/warehouse_060_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 31201 31720 -f1
/flx8/flat_061/warehouse_061_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 31721 32240 -f1
/flx8/flat_062/warehouse_062_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 32241 32760 -f1
/flx8/flat_063/warehouse_063_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 32761 33280 -f1
/flx8/flat_064/warehouse_064_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 33281 33800 -f1
/flx8/flat_065/warehouse_065_1.dat

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 33801 34320 -f1
/flx8/flat_066/warehouse_066_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 34321 34840 -f1
/flx8/flat_067/warehouse_067_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 34841 35360 -f1
/flx8/flat_068/warehouse_068_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 35361 35880 -f1
/flx8/flat_069/warehouse_069_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 35881 36400 -f1
/flx8/flat_070/warehouse_070_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 36401 36920 -f1
/flx8/flat_071/warehouse_071_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 36921 37440 -f1
/flx8/flat_072/warehouse_072_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 37441 37960 -f1
/flx8/flat_073/warehouse_073_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 37961 38480 -f1
/flx8/flat_074/warehouse_074_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 38481 39000 -f1
/flx8/flat_075/warehouse_075_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 39001 39520 -f1
/flx8/flat_076/warehouse_076_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 39521 40040 -f1
/flx8/flat_077/warehouse_077_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 40041 40560 -f1
/flx8/flat_078/warehouse_078_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 40561 41080 -f1
/flx8/flat_079/warehouse_079_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 41081 41600 -f1
/flx8/flat_080/warehouse_080_1.dat
```

DDL/LOAD_CUSTOMER_ALL.ddl

```
 CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_001/customer_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_002/customer_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_003/customer_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_004/customer_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_005/customer_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_006/customer_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_007/customer_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_008/customer_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_009/customer_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_010/customer_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_011/customer_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_012/customer_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_013/customer_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_014/customer_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_015/customer_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_016/customer_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_017/customer_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_018/customer_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_019/customer_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_020/customer_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_021/customer_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_022/customer_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER22;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_023/customer_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_024/customer_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_025/customer_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_026/customer_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_027/customer_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_028/customer_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_029/customer_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_030/customer_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_031/customer_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_032/customer_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_033/customer_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_034/customer_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_035/customer_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_036/customer_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_037/customer_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_038/customer_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_039/customer_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER40 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_040/customer_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER41 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_041/customer_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER42 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_042/customer_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER43 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_043/customer_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER44 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_044/customer_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER45 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_045/customer_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER45;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER46 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_046/customer_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER47 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_047/customer_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER48 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_048/customer_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER48;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER49 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_049/customer_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER50 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_050/customer_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER51 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_051/customer_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER52 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_052/customer_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER53 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_053/customer_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER54 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_054/customer_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER55 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_055/customer_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER56 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_056/customer_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER57 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_057/customer_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER58 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_058/customer_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER59 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_059/customer_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER60 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_060/customer_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER61 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_061/customer_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER61;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER62 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_062/customer_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER63 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_063/customer_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER64 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_064/customer_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER65 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_065/customer_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER66 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_066/customer_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER67 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_067/customer_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER68 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_068/customer_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER69 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_069/customer_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER70 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_070/customer_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER71 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_071/customer_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER71;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER72 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_072/customer_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER73 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_073/customer_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER73;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER74 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_074/customer_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER74;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER75 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_075/customer_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER76 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_076/customer_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER77 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_077/customer_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER78 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_078/customer_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER79 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_079/customer_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER80 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_080/customer_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 15600000 INSERT INTO CUSTOMER80;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_DISTRICT_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_001/district_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_002/district_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_003/district_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_004/district_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_005/district_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_006/district_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_007/district_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_008/district_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_009/district_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_010/district_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_011/district_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_012/district_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_013/district_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_014/district_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_015/district_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_016/district_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_017/district_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_018/district_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_019/district_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_020/district_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_021/district_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_022/district_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_023/district_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_024/district_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_025/district_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_026/district_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
IMPORT FROM /flx8/flat_027/district_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_028/district_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_029/district_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_030/district_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_031/district_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_032/district_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_033/district_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_034/district_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_035/district_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_036/district_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_037/district_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_038/district_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_039/district_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_040/district_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_041/district_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_042/district_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_043/district_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_044/district_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_045/district_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT45;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_046/district_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_047/district_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_048/district_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_049/district_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_050/district_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_051/district_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_052/district_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_053/district_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_054/district_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_055/district_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_056/district_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_057/district_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_058/district_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_059/district_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_060/district_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_061/district_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT61;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_062/district_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_063/district_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_064/district_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_065/district_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_066/district_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_067/district_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_068/district_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_069/district_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_070/district_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_071/district_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT71;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_072/district_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_073/district_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT73;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_074/district_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT74;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_075/district_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_076/district_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_077/district_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_078/district_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_079/district_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_080/district_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO DISTRICT80;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_HISTORY_ALL.ddl

```
connect to TPCC in share mode;
LOAD FROM /flx8/flat_001/history_001_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY1 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_002/history_002_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY2 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_003/history_003_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY3 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_004/history_004_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY4 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_005/history_005_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY5 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_006/history_006_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY6 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_007/history_007_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY7 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_008/history_008_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY8 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_009/history_009_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY9 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_010/history_010_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY10 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_011/history_011_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY11 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_012/history_012_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY12 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_013/history_013_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY13 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_014/history_014_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY14 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_015/history_015_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY15 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_016/history_016_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY16 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_017/history_017_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY17 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_018/history_018_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY18 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
```

```
connect to TPCC in share mode;
LOAD FROM /flx8/flat_019/history_019_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY19 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_020/history_020_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY20 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_021/history_021_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY21 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_022/history_022_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY22 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_023/history_023_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY23 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_024/history_024_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY24 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_025/history_025_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY25 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_026/history_026_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY26 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_027/history_027_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY27 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_028/history_028_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY28 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_029/history_029_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY29 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_030/history_030_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY30 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_031/history_031_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY31 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_032/history_032_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY32 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
```

```
connect to TPCC in share mode;
LOAD FROM /flx8/flat_033/history_033_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY33 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_034/history_034_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY34 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_035/history_035_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY35 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_036/history_036_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY36 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_037/history_037_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY37 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_038/history_038_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY38 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_039/history_039_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY39 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_040/history_040_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY40 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_041/history_041_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY41 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_042/history_042_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY42 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_043/history_043_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY43 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_044/history_044_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY44 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_045/history_045_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY45 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_046/history_046_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY46 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
```

```
connect to TPCC in share mode;
LOAD FROM /flx8/flat_047/history_047_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY47 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_048/history_048_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY48 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_049/history_049_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY49 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_050/history_050_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY50 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_051/history_051_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY51 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_052/history_052_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY52 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_053/history_053_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY53 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_054/history_054_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY54 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_055/history_055_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY55 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_056/history_056_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY56 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_057/history_057_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY57 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_058/history_058_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY58 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_059/history_059_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY59 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_060/history_060_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY60 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
```

```
connect to TPCC in share mode;
LOAD FROM /flx8/flat_061/history_061_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY61 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_062/history_062_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY62 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_063/history_063_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY63 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_064/history_064_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY64 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_065/history_065_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY65 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_066/history_066_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY66 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_067/history_067_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY67 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_068/history_068_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY68 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_069/history_069_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY69 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_070/history_070_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY70 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_071/history_071_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY71 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_072/history_072_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY72 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_073/history_073_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY73 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_074/history_074_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY74 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_075/history_075_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY75 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_076/history_076_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY76 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_077/history_077_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY77 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_078/history_078_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY78 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_079/history_079_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY79 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flx8/flat_080/history_080_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO HISTORY80 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 4 ;
connect reset;
```

DDL/LOAD_ITEM_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat/item_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1000 INSERT INTO ITEM;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_NEW_ORDER_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_001/neworder_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_002/neworder_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_003/neworder_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_004/neworder_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_005/neworder_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_006/neworder_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_007/neworder_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_008/neworder_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_009/neworder_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_010/neworder_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_011/neworder_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_012/neworder_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_013/neworder_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_014/neworder_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_015/neworder_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_016/neworder_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA16;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_017/neworder_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_018/neworder_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_019/neworder_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_020/neworder_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_021/neworder_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_022/neworder_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_023/neworder_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_024/neworder_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_025/neworder_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_026/neworder_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_027/neworder_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_028/neworder_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_029/neworder_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_030/neworder_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_031/neworder_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_032/neworder_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_033/neworder_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_034/neworder_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_035/neworder_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_036/neworder_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_037/neworder_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_038/neworder_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_039/neworder_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_040/neworder_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_041/neworder_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_042/neworder_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_043/neworder_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_044/neworder_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_045/neworder_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA45;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_046/neworder_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_047/neworder_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_048/neworder_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_049/neworder_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_050/neworder_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
IMPORT FROM /flx8/flat_051/neworder_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_052/neworder_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_053/neworder_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_054/neworder_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_055/neworder_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_056/neworder_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_057/neworder_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_058/neworder_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_059/neworder_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_060/neworder_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_061/neworder_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA61;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_062/neworder_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_063/neworder_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_064/neworder_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_065/neworder_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_066/neworder_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_067/neworder_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_068/neworder_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_069/neworder_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_070/neworder_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_071/neworder_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA71;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_072/neworder_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_073/neworder_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA73;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_074/neworder_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA74;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_075/neworder_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_076/neworder_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_077/neworder_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_078/neworder_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_079/neworder_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_080/neworder_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 20000 INSERT INTO NEW_ORDERA80;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_ORDERS_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_001/orders_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_002/orders_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_003/orders_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_004/orders_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
IMPORT FROM /flx8/flat_005/orders_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_006/orders_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_007/orders_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_008/orders_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_009/orders_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_010/orders_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_011/orders_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_012/orders_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_013/orders_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_014/orders_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_015/orders_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_016/orders_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_017/orders_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_018/orders_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_019/orders_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_020/orders_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_021/orders_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_022/orders_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_023/orders_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_024/orders_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_025/orders_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_026/orders_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_027/orders_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_028/orders_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_029/orders_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_030/orders_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_031/orders_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_032/orders_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_033/orders_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_034/orders_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_035/orders_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_036/orders_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_037/orders_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_038/orders_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_039/orders_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS39;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_040/orders_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_041/orders_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_042/orders_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_043/orders_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_044/orders_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_045/orders_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS45;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_046/orders_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_047/orders_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_048/orders_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_049/orders_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_050/orders_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_051/orders_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_052/orders_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_053/orders_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_054/orders_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_055/orders_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_056/orders_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_057/orders_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_058/orders_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_059/orders_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_060/orders_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_061/orders_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS61;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_062/orders_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_063/orders_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_064/orders_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_065/orders_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_066/orders_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_067/orders_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_068/orders_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_069/orders_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_070/orders_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_071/orders_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS71;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_072/orders_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_073/orders_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS73;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
IMPORT FROM /flx8/flat_074/orders_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS74;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_075/orders_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_076/orders_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_077/orders_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_078/orders_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_079/orders_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_080/orders_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO ORDERS80;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_ORDER_LINE_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_001/orderline_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_002/orderline_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_003/orderline_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_004/orderline_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_005/orderline_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_006/orderline_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_007/orderline_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_008/orderline_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_009/orderline_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_010/orderline_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_011/orderline_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_012/orderline_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_013/orderline_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_014/orderline_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_015/orderline_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_016/orderline_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_017/orderline_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_018/orderline_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_019/orderline_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_020/orderline_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_021/orderline_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE21;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_022/orderline_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_023/orderline_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_024/orderline_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_025/orderline_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_026/orderline_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_027/orderline_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_028/orderline_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_029/orderline_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_030/orderline_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_031/orderline_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_032/orderline_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_033/orderline_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_034/orderline_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_035/orderline_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_036/orderline_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_037/orderline_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_038/orderline_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_039/orderline_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE40 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_040/orderline_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE41 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_041/orderline_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE42 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_042/orderline_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE43 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_043/orderline_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE44 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_044/orderline_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE45 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_045/orderline_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE45;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE46 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_046/orderline_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE47 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_047/orderline_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE47;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE48 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_048/orderline_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE49 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_049/orderline_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE50 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_050/orderline_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE51 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_051/orderline_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE52 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_052/orderline_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE53 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_053/orderline_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE54 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_054/orderline_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE55 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_055/orderline_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE56 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_056/orderline_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE57 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_057/orderline_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE58 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_058/orderline_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE59 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_059/orderline_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE60 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_060/orderline_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE61 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_061/orderline_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE61;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE62 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_062/orderline_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE63 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_063/orderline_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE64 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_064/orderline_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE65 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_065/orderline_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE66 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_066/orderline_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE67 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_067/orderline_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE68 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_068/orderline_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE69 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_069/orderline_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE70 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_070/orderline_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE71 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_071/orderline_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE71;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE72 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_072/orderline_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE73 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_073/orderline_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE73;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE74 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_074/orderline_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE74;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE75 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_075/orderline_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE76 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_076/orderline_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE77 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_077/orderline_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE78 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_078/orderline_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE79 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_079/orderline_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE80 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_080/orderline_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 171600000 INSERT INTO ORDER_LINE80;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_STOCK_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_001/stock_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_002/stock_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_003/stock_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_004/stock_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_005/stock_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_006/stock_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_007/stock_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_008/stock_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_009/stock_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_010/stock_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_011/stock_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_012/stock_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_013/stock_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_014/stock_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_015/stock_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_016/stock_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_017/stock_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_018/stock_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_019/stock_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK19;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_020/stock_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_021/stock_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_022/stock_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_023/stock_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_024/stock_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_025/stock_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_026/stock_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_027/stock_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_028/stock_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_029/stock_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_030/stock_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_031/stock_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_032/stock_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_033/stock_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_034/stock_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_035/stock_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_036/stock_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_037/stock_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_038/stock_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_039/stock_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK40 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_040/stock_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK41 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_041/stock_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK42 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_042/stock_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK43 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_043/stock_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK44 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_044/stock_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK45 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_045/stock_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK45;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK46 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_046/stock_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK47 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_047/stock_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK48 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_048/stock_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK49 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_049/stock_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK50 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_050/stock_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK51 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_051/stock_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK52 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_052/stock_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK53 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_053/stock_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK54 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_054/stock_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK55 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_055/stock_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK56 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_056/stock_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK57 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_057/stock_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK58 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_058/stock_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK59 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_059/stock_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK60 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_060/stock_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK61 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_061/stock_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK61;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK62 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_062/stock_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK63 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_063/stock_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK64 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_064/stock_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK65 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_065/stock_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK66 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_066/stock_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK67 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_067/stock_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK68 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_068/stock_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK69 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_069/stock_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK70 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_070/stock_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK71 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_071/stock_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK71;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK72 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_072/stock_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK73 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_073/stock_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK73;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK74 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_074/stock_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK74;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK75 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_075/stock_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK76 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_076/stock_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK77 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_077/stock_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK78 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_078/stock_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK79 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_079/stock_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK80 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_080/stock_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 52000000 INSERT INTO STOCK80;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_WAREHOUSE_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_001/warehouse_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_002/warehouse_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_003/warehouse_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_004/warehouse_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_005/warehouse_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_006/warehouse_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_007/warehouse_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_008/warehouse_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_009/warehouse_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_010/warehouse_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_011/warehouse_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_012/warehouse_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_013/warehouse_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_014/warehouse_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_015/warehouse_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_016/warehouse_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_017/warehouse_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_018/warehouse_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_019/warehouse_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_020/warehouse_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_021/warehouse_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_022/warehouse_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE22;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_023/warehouse_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_024/warehouse_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_025/warehouse_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_026/warehouse_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_027/warehouse_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_028/warehouse_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_029/warehouse_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_030/warehouse_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_031/warehouse_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_032/warehouse_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_033/warehouse_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_034/warehouse_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_035/warehouse_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_036/warehouse_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_037/warehouse_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_038/warehouse_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_039/warehouse_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_040/warehouse_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_041/warehouse_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_042/warehouse_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_043/warehouse_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_044/warehouse_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_045/warehouse_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE45;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_046/warehouse_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_047/warehouse_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_048/warehouse_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_049/warehouse_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_050/warehouse_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_051/warehouse_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_052/warehouse_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_053/warehouse_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_054/warehouse_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_055/warehouse_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_056/warehouse_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
IMPORT FROM /flx8/flat_057/warehouse_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_058/warehouse_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_059/warehouse_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_060/warehouse_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_061/warehouse_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE61;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_062/warehouse_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_063/warehouse_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_064/warehouse_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_065/warehouse_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_066/warehouse_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_067/warehouse_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_068/warehouse_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_069/warehouse_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_070/warehouse_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_071/warehouse_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE71;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_072/warehouse_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_073/warehouse_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE73;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_074/warehouse_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE74;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_075/warehouse_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_076/warehouse_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_077/warehouse_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_078/warehouse_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_079/warehouse_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_080/warehouse_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE80;
COMMIT WORK;
CONNECT RESET;
```

DDL/RNST_CUSTOMER.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_001/warehouse_001_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_002/warehouse_002_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_003/warehouse_003_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_004/warehouse_004_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_005/warehouse_005_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_006/warehouse_006_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_007/warehouse_007_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_008/warehouse_008_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_009/warehouse_009_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_010/warehouse_010_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
IMPORT FROM /flx8/flat_011/warehouse_011_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_012/warehouse_012_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_013/warehouse_013_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_014/warehouse_014_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_015/warehouse_015_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_016/warehouse_016_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_017/warehouse_017_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_018/warehouse_018_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_019/warehouse_019_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_020/warehouse_020_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_021/warehouse_021_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_022/warehouse_022_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_023/warehouse_023_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_024/warehouse_024_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_025/warehouse_025_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_026/warehouse_026_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_027/warehouse_027_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_028/warehouse_028_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_029/warehouse_029_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_030/warehouse_030_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_031/warehouse_031_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_032/warehouse_032_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_033/warehouse_033_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_034/warehouse_034_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_035/warehouse_035_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_036/warehouse_036_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_037/warehouse_037_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_038/warehouse_038_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_039/warehouse_039_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_040/warehouse_040_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE40;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_041/warehouse_041_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE41;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_042/warehouse_042_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE42;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_043/warehouse_043_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE43;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_044/warehouse_044_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE44;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_045/warehouse_045_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE45;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_046/warehouse_046_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE46;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_047/warehouse_047_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE47;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_048/warehouse_048_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE48;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_049/warehouse_049_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE49;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_050/warehouse_050_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE50;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_051/warehouse_051_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE51;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_052/warehouse_052_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE52;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_053/warehouse_053_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE53;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_054/warehouse_054_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE54;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_055/warehouse_055_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE55;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_056/warehouse_056_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE56;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_057/warehouse_057_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE57;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_058/warehouse_058_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE58;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_059/warehouse_059_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE59;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_060/warehouse_060_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE60;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_061/warehouse_061_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE61;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_062/warehouse_062_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE62;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_063/warehouse_063_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE63;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_064/warehouse_064_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE64;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_065/warehouse_065_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE65;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_066/warehouse_066_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE66;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_067/warehouse_067_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE67;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_068/warehouse_068_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE68;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_069/warehouse_069_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE69;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_070/warehouse_070_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE70;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_071/warehouse_071_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE71;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_072/warehouse_072_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE72;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_073/warehouse_073_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE73;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_074/warehouse_074_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE74;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_075/warehouse_075_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE75;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_076/warehouse_076_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE76;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_077/warehouse_077_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE77;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_078/warehouse_078_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE78;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flx8/flat_079/warehouse_079_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE79;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
IMPORT FROM /flx8/flat_080/warehouse_080_1.dat OF DEL MODIFIED BY COLDEL|
TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO WAREHOUSE80;
COMMIT WORK;
CONNECT RESET;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
```

```
RUNSTATS ON TABLE tpcc.CUSTOMER53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_DISTRICT.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT77 AND INDEXES ALL;
```

```
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_HISTORY.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
```

```
RUNSTATS ON TABLE tpcc.HISTORY49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_ITEM.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ITEM AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_NEW_ORDERA.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA71 AND INDEXES ALL;
```

```
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_NEW_ORDERB.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
```

```
RUNSTATS ON TABLE tpcc.NEW_ORDERB43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_ORDERS.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS67 AND INDEXES ALL;
```

```
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_ORDER_LINE.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_STOCK.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK10 AND INDEXES ALL;
COMMIT WORK;
```

```
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
```

```
RUNSTATS ON TABLE tpcc.STOCK63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_WAREHOUSE.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK39 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK40 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK41 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK42 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK43 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK44 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK45 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK46 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK47 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK48 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK49 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK50 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK51 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK52 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK53 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK54 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK55 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK56 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK57 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK58 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK59 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK60 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK61 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK62 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK63 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK64 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK65 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK66 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK67 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK68 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK69 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK70 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK71 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK72 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK73 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK74 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK75 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK76 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK77 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK78 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK79 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK80 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

README

Database DDL
==========

No shcema changes.

DDL generated for 48 UNION-ALL arms on all tables, except ITEM.

bp/create_bufferpool.ddl

```
connect to tpcc;

CREATE BUFFERPOOL "IBMDEFAULT8K"  SIZE 1000 PAGESIZE 8192 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "IBMDEFAULT16K"  SIZE 1000 PAGESIZE 16384 NOT
EXTENDED STORAGE;

connect reset;
terminate;
```

db/create_database.ddl

```
drop database tpcc;
create database tpcc on /db_home collate using identity
catalog tablespace
managed by system using ('/db_home/db1catalog');
```

ts/create_tablespace.ddl

```
connect to tpcc;

drop tablespace ts_wh_001;
create regular tablespace ts_wh_001 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw1' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_002;
create regular tablespace ts_wh_002 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw2' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_003;
create regular tablespace ts_wh_003 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw3' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_004;
create regular tablespace ts_wh_004 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw4' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_005;
create regular tablespace ts_wh_005 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw5' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_006;
create regular tablespace ts_wh_006 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw6' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_007;
create regular tablespace ts_wh_007 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw7' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_008;
create regular tablespace ts_wh_008 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw8' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_009;
create regular tablespace ts_wh_009 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw9' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_010;
create regular tablespace ts_wh_010 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw10' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_011;
create regular tablespace ts_wh_011 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw11' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_012;
create regular tablespace ts_wh_012 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw12' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_013;
create regular tablespace ts_wh_013 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw13' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_014;
create regular tablespace ts_wh_014 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw14' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_015;
create regular tablespace ts_wh_015 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw15' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_016;
create regular tablespace ts_wh_016 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw16' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_017;
create regular tablespace ts_wh_017 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw17' 224
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_wh_018;
create regular tablespace ts_wh_018 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw18' 224
             )
       extentsize 32
```

```
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_019;
create regular tablespace ts_wh_019 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw19' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_020;
create regular tablespace ts_wh_020 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw20' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_021;
create regular tablespace ts_wh_021 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw21' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;

drop tablespace ts_wh_022;
create regular tablespace ts_wh_022 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw22' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_023;
create regular tablespace ts_wh_023 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw23' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_024;
create regular tablespace ts_wh_024 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw24' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_025;
create regular tablespace ts_wh_025 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw25' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_026;
create regular tablespace ts_wh_026 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw26' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_027;
create regular tablespace ts_wh_027 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw27' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_028;
create regular tablespace ts_wh_028 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw28' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_029;
create regular tablespace ts_wh_029 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw29' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_030;
create regular tablespace ts_wh_030 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw30' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_031;
create regular tablespace ts_wh_031 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw31' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_032;
create regular tablespace ts_wh_032 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw32' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_033;
create regular tablespace ts_wh_033 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw33' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_034;
create regular tablespace ts_wh_034 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw34' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_035;
create regular tablespace ts_wh_035 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw35' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_036;
create regular tablespace ts_wh_036 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw36' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_037;
create regular tablespace ts_wh_037 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw37' 224
               )
        extentsize 32
```

```
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_038;
create regular tablespace ts_wh_038 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw38' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_039;
create regular tablespace ts_wh_039 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw39' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_040;
create regular tablespace ts_wh_040 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw40' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;


drop tablespace ts_wh_041;
create regular tablespace ts_wh_041 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw41' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_042;
create regular tablespace ts_wh_042 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw42' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_043;
create regular tablespace ts_wh_043 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw43' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_044;
create regular tablespace ts_wh_044 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw44' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_045;
create regular tablespace ts_wh_045 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw45' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_046;
create regular tablespace ts_wh_046 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw46' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_047;
create regular tablespace ts_wh_047 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw47' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_048;
create regular tablespace ts_wh_048 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw48' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_049;
create regular tablespace ts_wh_049 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw49' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_050;
create regular tablespace ts_wh_050 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw50' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_051;
create regular tablespace ts_wh_051 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw51' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_052;
create regular tablespace ts_wh_052 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw52' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_053;
create regular tablespace ts_wh_053 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw53' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_054;
create regular tablespace ts_wh_054 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw54' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_055;
create regular tablespace ts_wh_055 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw55' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_056;
create regular tablespace ts_wh_056 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw56' 224
               )
```

```
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_057;
create regular tablespace ts_wh_057 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw57' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_058;
create regular tablespace ts_wh_058 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw58' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_059;
create regular tablespace ts_wh_059 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw59' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_060;
create regular tablespace ts_wh_060 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw60' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_061;
create regular tablespace ts_wh_061 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw61' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;

drop tablespace ts_wh_062;
create regular tablespace ts_wh_062 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw62' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_063;
create regular tablespace ts_wh_063 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw63' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_064;
create regular tablespace ts_wh_064 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw64' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_065;
create regular tablespace ts_wh_065 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw65' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_066;
create regular tablespace ts_wh_066 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw66' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_067;
create regular tablespace ts_wh_067 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw67' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_068;
create regular tablespace ts_wh_068 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw68' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_069;
create regular tablespace ts_wh_069 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw69' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_070;
create regular tablespace ts_wh_070 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw70' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_071;
create regular tablespace ts_wh_071 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw71' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_072;
create regular tablespace ts_wh_072 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw72' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_073;
create regular tablespace ts_wh_073 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw73' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_074;
create regular tablespace ts_wh_074 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw74' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_075;
create regular tablespace ts_wh_075 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw75' 224
               )
```

```
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_076;
create regular tablespace ts_wh_076 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw76' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_077;
create regular tablespace ts_wh_077 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw77' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_078;
create regular tablespace ts_wh_078 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw78' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_079;
create regular tablespace ts_wh_079 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw79' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_wh_080;
create regular tablespace ts_wh_080 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw80' 224
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;




drop tablespace ts_dis_001;
create regular tablespace ts_dis_001 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw81' 576
               )
```

```
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_002;
create regular tablespace ts_dis_002 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw82' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_003;
create regular tablespace ts_dis_003 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw83' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_004;
create regular tablespace ts_dis_004 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw84' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_005;
create regular tablespace ts_dis_005 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw85' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_006;
create regular tablespace ts_dis_006 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw86' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_007;
create regular tablespace ts_dis_007 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw87' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
```

```
drop tablespace ts_dis_008;
create regular tablespace ts_dis_008 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw88' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_009;
create regular tablespace ts_dis_009 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw89' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_010;
create regular tablespace ts_dis_010 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw90' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_011;
create regular tablespace ts_dis_011 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw91' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_012;
create regular tablespace ts_dis_012 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw92' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_013;
create regular tablespace ts_dis_013 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw93' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_014;
create regular tablespace ts_dis_014 pagesize 4K
         managed by database
         using
```

```
        (
                device '/dev/raw/raw94' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_015;
create regular tablespace ts_dis_015 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw95' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_016;
create regular tablespace ts_dis_016 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw96' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_017;
create regular tablespace ts_dis_017 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw97' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_018;
create regular tablespace ts_dis_018 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw98' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_019;
create regular tablespace ts_dis_019 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw99' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_020;
create regular tablespace ts_dis_020 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw100' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_021;
create regular tablespace ts_dis_021 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw101' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;

drop tablespace ts_dis_022;
create regular tablespace ts_dis_022 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw102' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_023;
create regular tablespace ts_dis_023 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw103' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_024;
create regular tablespace ts_dis_024 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw104' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_025;
create regular tablespace ts_dis_025 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw105' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_026;
create regular tablespace ts_dis_026 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw106' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_027;
create regular tablespace ts_dis_027 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw107' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_028;
create regular tablespace ts_dis_028 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw108' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_029;
create regular tablespace ts_dis_029 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw109' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_030;
create regular tablespace ts_dis_030 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw110' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_031;
create regular tablespace ts_dis_031 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw111' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_032;
create regular tablespace ts_dis_032 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw112' 576
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_dis_033;
create regular tablespace ts_dis_033 pagesize 4K
         managed by database
         using
```

```
      (
            device '/dev/raw/raw113' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_034;
create regular tablespace ts_dis_034 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw114' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_035;
create regular tablespace ts_dis_035 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw115' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_036;
create regular tablespace ts_dis_036 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw116' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_037;
create regular tablespace ts_dis_037 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw117' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_038;
create regular tablespace ts_dis_038 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw118' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_039;
create regular tablespace ts_dis_039 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw119' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_040;
create regular tablespace ts_dis_040 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw120' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;


drop tablespace ts_dis_041;
create regular tablespace ts_dis_041 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw121' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_042;
create regular tablespace ts_dis_042 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw122' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_043;
create regular tablespace ts_dis_043 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw123' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_044;
create regular tablespace ts_dis_044 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw124' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_045;
create regular tablespace ts_dis_045 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw125' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_046;
create regular tablespace ts_dis_046 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw126' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_047;
create regular tablespace ts_dis_047 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw127' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_048;
create regular tablespace ts_dis_048 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw128' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_049;
create regular tablespace ts_dis_049 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw129' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_050;
create regular tablespace ts_dis_050 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw130' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_051;
create regular tablespace ts_dis_051 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw131' 576
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_dis_052;
create regular tablespace ts_dis_052 pagesize 4K
       managed by database
```

```
        using
       (
              device '/dev/raw/raw132' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_053;
create regular tablespace ts_dis_053 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw133' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_054;
create regular tablespace ts_dis_054 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw134' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_055;
create regular tablespace ts_dis_055 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw135' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_056;
create regular tablespace ts_dis_056 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw136' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_057;
create regular tablespace ts_dis_057 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw137' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_058;
create regular tablespace ts_dis_058 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw138' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_059;
create regular tablespace ts_dis_059 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw139' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_060;
create regular tablespace ts_dis_060 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw140' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_061;
create regular tablespace ts_dis_061 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw141' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;

drop tablespace ts_dis_062;
create regular tablespace ts_dis_062 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw142' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_063;
create regular tablespace ts_dis_063 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw143' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_064;
create regular tablespace ts_dis_064 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw144' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_065;
create regular tablespace ts_dis_065 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw145' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_066;
create regular tablespace ts_dis_066 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw146' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_067;
create regular tablespace ts_dis_067 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw147' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_068;
create regular tablespace ts_dis_068 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw148' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_069;
create regular tablespace ts_dis_069 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw149' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_070;
create regular tablespace ts_dis_070 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw150' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_071;
create regular tablespace ts_dis_071 pagesize 4K
        managed by database
```

```
        using
       (
              device '/dev/raw/raw151' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_072;
create regular tablespace ts_dis_072 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw152' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_073;
create regular tablespace ts_dis_073 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw153' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_074;
create regular tablespace ts_dis_074 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw154' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_075;
create regular tablespace ts_dis_075 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw155' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_076;
create regular tablespace ts_dis_076 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw156' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_077;
create regular tablespace ts_dis_077 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw157' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_078;
create regular tablespace ts_dis_078 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw158' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_079;
create regular tablespace ts_dis_079 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw159' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_dis_080;
create regular tablespace ts_dis_080 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw160' 576
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
```

```
drop tablespace ts_item;
create regular tablespace ts_item pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw161' 48,
              device '/dev/raw/raw162' 48,
              device '/dev/raw/raw163' 48,
              device '/dev/raw/raw164' 48,
              device '/dev/raw/raw165' 48,
              device '/dev/raw/raw166' 48,
              device '/dev/raw/raw167' 48,
              device '/dev/raw/raw168' 48,
              device '/dev/raw/raw169' 48,
              device '/dev/raw/raw170' 48,
              device '/dev/raw/raw171' 48,
              device '/dev/raw/raw172' 48,
              device '/dev/raw/raw173' 48,
              device '/dev/raw/raw174' 48,
              device '/dev/raw/raw175' 48,
              device '/dev/raw/raw176' 48,
              device '/dev/raw/raw177' 48,
              device '/dev/raw/raw178' 48,
              device '/dev/raw/raw179' 48,
              device '/dev/raw/raw180' 48,
              device '/dev/raw/raw181' 48,
              device '/dev/raw/raw182' 48,
              device '/dev/raw/raw183' 48,
              device '/dev/raw/raw184' 48,
              device '/dev/raw/raw185' 48,
              device '/dev/raw/raw186' 48,
              device '/dev/raw/raw187' 48,
              device '/dev/raw/raw188' 48,
              device '/dev/raw/raw189' 48,
              device '/dev/raw/raw190' 48,
              device '/dev/raw/raw191' 48,
              device '/dev/raw/raw192' 48,
              device '/dev/raw/raw193' 48,
              device '/dev/raw/raw194' 48,
              device '/dev/raw/raw195' 48,
              device '/dev/raw/raw196' 48,
              device '/dev/raw/raw197' 48,
              device '/dev/raw/raw198' 48,
              device '/dev/raw/raw199' 48,
              device '/dev/raw/raw200' 48,
              device '/dev/raw/raw201' 48,
              device '/dev/raw/raw202' 48,
              device '/dev/raw/raw203' 48,
              device '/dev/raw/raw204' 48,
              device '/dev/raw/raw205' 48,
              device '/dev/raw/raw206' 48,
              device '/dev/raw/raw207' 48,
              device '/dev/raw/raw208' 48,
              device '/dev/raw/raw209' 48,
              device '/dev/raw/raw210' 48,
              device '/dev/raw/raw211' 48,
              device '/dev/raw/raw212' 48,
              device '/dev/raw/raw213' 48,
              device '/dev/raw/raw214' 48,
              device '/dev/raw/raw215' 48,
              device '/dev/raw/raw216' 48,
              device '/dev/raw/raw217' 48,
              device '/dev/raw/raw218' 48,
              device '/dev/raw/raw219' 48,
              device '/dev/raw/raw220' 48,
              device '/dev/raw/raw221' 48,
              device '/dev/raw/raw222' 48,
              device '/dev/raw/raw223' 48,
              device '/dev/raw/raw224' 48,
              device '/dev/raw/raw225' 48,
              device '/dev/raw/raw226' 48,
              device '/dev/raw/raw227' 48,
              device '/dev/raw/raw228' 48,
              device '/dev/raw/raw229' 48,
              device '/dev/raw/raw230' 48,
              device '/dev/raw/raw231' 48,
              device '/dev/raw/raw232' 48,
              device '/dev/raw/raw233' 48,
              device '/dev/raw/raw234' 48,
              device '/dev/raw/raw235' 48,
              device '/dev/raw/raw236' 48,
              device '/dev/raw/raw237' 48,
              device '/dev/raw/raw238' 48,
              device '/dev/raw/raw239' 48,
              device '/dev/raw/raw240' 48
             )
       extentsize 16
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;


drop tablespace ts_stock_001;
create regular tablespace ts_stock_001 pagesize 4K
        managed by database
        using
       (
```

```
            device '/dev/raw/raw241' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_002;
create regular tablespace ts_stock_002 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw242' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_003;
create regular tablespace ts_stock_003 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw243' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_004;
create regular tablespace ts_stock_004 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw244' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_005;
create regular tablespace ts_stock_005 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw245' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_006;
create regular tablespace ts_stock_006 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw246' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_007;
create regular tablespace ts_stock_007 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw247' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_008;
create regular tablespace ts_stock_008 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw248' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_009;
create regular tablespace ts_stock_009 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw249' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_010;
create regular tablespace ts_stock_010 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw250' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_011;
create regular tablespace ts_stock_011 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw251' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_012;
create regular tablespace ts_stock_012 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw252' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_013;
create regular tablespace ts_stock_013 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw253' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_014;
create regular tablespace ts_stock_014 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw254' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_015;
create regular tablespace ts_stock_015 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw255' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_016;
create regular tablespace ts_stock_016 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw256' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_017;
create regular tablespace ts_stock_017 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw257' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_018;
create regular tablespace ts_stock_018 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw258' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_019;
create regular tablespace ts_stock_019 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw259' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_020;
create regular tablespace ts_stock_020 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw260' 4551680
```

```
            )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_021;
create regular tablespace ts_stock_021 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw261' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;

drop tablespace ts_stock_022;
create regular tablespace ts_stock_022 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw262' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_023;
create regular tablespace ts_stock_023 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw263' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_024;
create regular tablespace ts_stock_024 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw264' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_025;
create regular tablespace ts_stock_025 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw265' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_026;
create regular tablespace ts_stock_026 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw266' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_027;
create regular tablespace ts_stock_027 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw267' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_028;
create regular tablespace ts_stock_028 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw268' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_029;
create regular tablespace ts_stock_029 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw269' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_030;
create regular tablespace ts_stock_030 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw270' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_031;
create regular tablespace ts_stock_031 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw271' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_032;
create regular tablespace ts_stock_032 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw272' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_033;
create regular tablespace ts_stock_033 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw273' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_034;
create regular tablespace ts_stock_034 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw274' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_035;
create regular tablespace ts_stock_035 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw275' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_036;
create regular tablespace ts_stock_036 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw276' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_037;
create regular tablespace ts_stock_037 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw277' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_038;
create regular tablespace ts_stock_038 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw278' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_039;
create regular tablespace ts_stock_039 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw279' 4551680
```

```
            )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_040;
create regular tablespace ts_stock_040 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw280' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;


drop tablespace ts_stock_041;
create regular tablespace ts_stock_041 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw281' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_042;
create regular tablespace ts_stock_042 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw282' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_043;
create regular tablespace ts_stock_043 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw283' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_044;
create regular tablespace ts_stock_044 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw284' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_045;
create regular tablespace ts_stock_045 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw285' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_046;
create regular tablespace ts_stock_046 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw286' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_047;
create regular tablespace ts_stock_047 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw287' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_048;
create regular tablespace ts_stock_048 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw288' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_049;
create regular tablespace ts_stock_049 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw289' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_050;
create regular tablespace ts_stock_050 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw290' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_051;
create regular tablespace ts_stock_051 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw291' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_052;
create regular tablespace ts_stock_052 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw292' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_053;
create regular tablespace ts_stock_053 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw293' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_054;
create regular tablespace ts_stock_054 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw294' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_055;
create regular tablespace ts_stock_055 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw295' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_056;
create regular tablespace ts_stock_056 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw296' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_057;
create regular tablespace ts_stock_057 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw297' 4551680
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_stock_058;
create regular tablespace ts_stock_058 pagesize 4K
         managed by database
         using
        (
```

```
            device '/dev/raw/raw298' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_059;
create regular tablespace ts_stock_059 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw299' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_060;
create regular tablespace ts_stock_060 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw300' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_061;
create regular tablespace ts_stock_061 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw301' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;

drop tablespace ts_stock_062;
create regular tablespace ts_stock_062 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw302' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_063;
create regular tablespace ts_stock_063 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw303' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_064;
create regular tablespace ts_stock_064 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw304' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_065;
create regular tablespace ts_stock_065 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw305' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_066;
create regular tablespace ts_stock_066 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw306' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_067;
create regular tablespace ts_stock_067 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw307' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_068;
create regular tablespace ts_stock_068 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw308' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_069;
create regular tablespace ts_stock_069 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw309' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_070;
create regular tablespace ts_stock_070 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw310' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_071;
create regular tablespace ts_stock_071 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw311' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_072;
create regular tablespace ts_stock_072 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw312' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_073;
create regular tablespace ts_stock_073 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw313' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_074;
create regular tablespace ts_stock_074 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw314' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_075;
create regular tablespace ts_stock_075 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw315' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_076;
create regular tablespace ts_stock_076 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw316' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_077;
create regular tablespace ts_stock_077 pagesize 4K
       managed by database
       using
      (
```

```
            device '/dev/raw/raw317' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_078;
create regular tablespace ts_stock_078 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw318' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_079;
create regular tablespace ts_stock_079 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw319' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_stock_080;
create regular tablespace ts_stock_080 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw320' 4551680
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;



drop tablespace ts_customer_001;
create regular tablespace ts_customer_001 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw321' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_002;
create regular tablespace ts_customer_002 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw322' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_003;
create regular tablespace ts_customer_003 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw323' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_004;
create regular tablespace ts_customer_004 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw324' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_005;
create regular tablespace ts_customer_005 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw325' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_006;
create regular tablespace ts_customer_006 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw326' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_007;
create regular tablespace ts_customer_007 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw327' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_008;
create regular tablespace ts_customer_008 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw328' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_009;
create regular tablespace ts_customer_009 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw329' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_010;
create regular tablespace ts_customer_010 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw330' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_011;
create regular tablespace ts_customer_011 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw331' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_012;
create regular tablespace ts_customer_012 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw332' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_013;
create regular tablespace ts_customer_013 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw333' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_014;
create regular tablespace ts_customer_014 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw334' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_015;
create regular tablespace ts_customer_015 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw335' 3277568
           )
      extentsize 32
      bufferpool IBMDEFAULTBP
      prefetchsize 4096;
commit;
drop tablespace ts_customer_016;
create regular tablespace ts_customer_016 pagesize 4K
       managed by database
```

```
        using
        (
                device '/dev/raw/raw336' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_017;
create regular tablespace ts_customer_017 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw337' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_018;
create regular tablespace ts_customer_018 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw338' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_019;
create regular tablespace ts_customer_019 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw339' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_020;
create regular tablespace ts_customer_020 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw340' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_021;
create regular tablespace ts_customer_021 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw341' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;

drop tablespace ts_customer_022;
create regular tablespace ts_customer_022 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw342' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_023;
create regular tablespace ts_customer_023 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw343' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_024;
create regular tablespace ts_customer_024 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw344' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_025;
create regular tablespace ts_customer_025 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw345' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_026;
create regular tablespace ts_customer_026 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw346' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_027;
create regular tablespace ts_customer_027 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw347' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_028;
create regular tablespace ts_customer_028 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw348' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_029;
create regular tablespace ts_customer_029 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw349' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_030;
create regular tablespace ts_customer_030 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw350' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_031;
create regular tablespace ts_customer_031 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw351' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_032;
create regular tablespace ts_customer_032 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw352' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_033;
create regular tablespace ts_customer_033 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw353' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_034;
create regular tablespace ts_customer_034 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw354' 3277568
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_customer_035;
create regular tablespace ts_customer_035 pagesize 4K
         managed by database
```

```
 using
(
        device '/dev/raw/raw355' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_036;
create regular tablespace ts_customer_036 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw356' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_037;
create regular tablespace ts_customer_037 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw357' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_038;
create regular tablespace ts_customer_038 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw358' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_039;
create regular tablespace ts_customer_039 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw359' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_040;
create regular tablespace ts_customer_040 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw360' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;


drop tablespace ts_customer_041;
create regular tablespace ts_customer_041 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw361' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_042;
create regular tablespace ts_customer_042 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw362' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_043;
create regular tablespace ts_customer_043 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw363' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_044;
create regular tablespace ts_customer_044 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw364' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_045;
create regular tablespace ts_customer_045 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw365' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_046;
create regular tablespace ts_customer_046 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw366' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_047;
create regular tablespace ts_customer_047 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw367' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_048;
create regular tablespace ts_customer_048 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw368' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_049;
create regular tablespace ts_customer_049 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw369' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_050;
create regular tablespace ts_customer_050 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw370' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_051;
create regular tablespace ts_customer_051 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw371' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_052;
create regular tablespace ts_customer_052 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw372' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_053;
create regular tablespace ts_customer_053 pagesize 4K
 managed by database
 using
(
        device '/dev/raw/raw373' 3277568
       )
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_054;
create regular tablespace ts_customer_054 pagesize 4K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw374' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_055;
create regular tablespace ts_customer_055 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw375' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_056;
create regular tablespace ts_customer_056 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw376' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_057;
create regular tablespace ts_customer_057 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw377' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_058;
create regular tablespace ts_customer_058 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw378' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_059;
create regular tablespace ts_customer_059 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw379' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_060;
create regular tablespace ts_customer_060 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw380' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_061;
create regular tablespace ts_customer_061 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw381' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;

drop tablespace ts_customer_062;
create regular tablespace ts_customer_062 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw382' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_063;
create regular tablespace ts_customer_063 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw383' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_064;
create regular tablespace ts_customer_064 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw384' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_065;
create regular tablespace ts_customer_065 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw385' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_066;
create regular tablespace ts_customer_066 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw386' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_067;
create regular tablespace ts_customer_067 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw387' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_068;
create regular tablespace ts_customer_068 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw388' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_069;
create regular tablespace ts_customer_069 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw389' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_070;
create regular tablespace ts_customer_070 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw390' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_071;
create regular tablespace ts_customer_071 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw391' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_072;
create regular tablespace ts_customer_072 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw392' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_073;
create regular tablespace ts_customer_073 pagesize 4K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw393' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_074;
create regular tablespace ts_customer_074 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw394' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_075;
create regular tablespace ts_customer_075 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw395' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_076;
create regular tablespace ts_customer_076 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw396' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_077;
create regular tablespace ts_customer_077 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw397' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_078;
create regular tablespace ts_customer_078 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw398' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_079;
create regular tablespace ts_customer_079 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw399' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_customer_080;
create regular tablespace ts_customer_080 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw400' 3277568
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;



drop tablespace is_customer_001;
create regular tablespace is_customer_001 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw401' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_002;
create regular tablespace is_customer_002 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw402' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_003;
create regular tablespace is_customer_003 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw403' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_004;
create regular tablespace is_customer_004 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw404' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_005;
create regular tablespace is_customer_005 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw405' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_006;
create regular tablespace is_customer_006 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw406' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_007;
create regular tablespace is_customer_007 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw407' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_008;
create regular tablespace is_customer_008 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw408' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_009;
create regular tablespace is_customer_009 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw409' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_010;
create regular tablespace is_customer_010 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw410' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_011;
create regular tablespace is_customer_011 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw411' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
```

```
drop tablespace is_customer_012;
create regular tablespace is_customer_012 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw412' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_013;
create regular tablespace is_customer_013 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw413' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_014;
create regular tablespace is_customer_014 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw414' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_015;
create regular tablespace is_customer_015 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw415' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_016;
create regular tablespace is_customer_016 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw416' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_017;
create regular tablespace is_customer_017 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw417' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_018;
create regular tablespace is_customer_018 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw418' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_019;
create regular tablespace is_customer_019 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw419' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_020;
create regular tablespace is_customer_020 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw420' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_021;
create regular tablespace is_customer_021 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw421' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;

drop tablespace is_customer_022;
create regular tablespace is_customer_022 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw422' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_023;
create regular tablespace is_customer_023 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw423' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_024;
create regular tablespace is_customer_024 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw424' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_025;
create regular tablespace is_customer_025 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw425' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_026;
create regular tablespace is_customer_026 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw426' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_027;
create regular tablespace is_customer_027 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw427' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_028;
create regular tablespace is_customer_028 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw428' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_029;
create regular tablespace is_customer_029 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw429' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_030;
create regular tablespace is_customer_030 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw430' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
```

```
drop tablespace is_customer_031;
create regular tablespace is_customer_031 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw431' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_032;
create regular tablespace is_customer_032 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw432' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_033;
create regular tablespace is_customer_033 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw433' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_034;
create regular tablespace is_customer_034 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw434' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_035;
create regular tablespace is_customer_035 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw435' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_036;
create regular tablespace is_customer_036 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw436' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_037;
create regular tablespace is_customer_037 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw437' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_038;
create regular tablespace is_customer_038 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw438' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_039;
create regular tablespace is_customer_039 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw439' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_040;
create regular tablespace is_customer_040 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw440' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;


drop tablespace is_customer_041;
create regular tablespace is_customer_041 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw441' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_042;
create regular tablespace is_customer_042 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw442' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_043;
create regular tablespace is_customer_043 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw443' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_044;
create regular tablespace is_customer_044 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw444' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_045;
create regular tablespace is_customer_045 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw445' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_046;
create regular tablespace is_customer_046 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw446' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_047;
create regular tablespace is_customer_047 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw447' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_048;
create regular tablespace is_customer_048 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw448' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_049;
create regular tablespace is_customer_049 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw449' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
```

```
commit;
drop tablespace is_customer_050;
create regular tablespace is_customer_050 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw450' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_051;
create regular tablespace is_customer_051 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw451' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_052;
create regular tablespace is_customer_052 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw452' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_053;
create regular tablespace is_customer_053 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw453' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_054;
create regular tablespace is_customer_054 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw454' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_055;
create regular tablespace is_customer_055 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw455' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_056;
create regular tablespace is_customer_056 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw456' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_057;
create regular tablespace is_customer_057 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw457' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_058;
create regular tablespace is_customer_058 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw458' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_059;
create regular tablespace is_customer_059 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw459' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_060;
create regular tablespace is_customer_060 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw460' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_061;
create regular tablespace is_customer_061 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw461' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;

drop tablespace is_customer_062;
create regular tablespace is_customer_062 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw462' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_063;
create regular tablespace is_customer_063 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw463' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_064;
create regular tablespace is_customer_064 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw464' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_065;
create regular tablespace is_customer_065 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw465' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_066;
create regular tablespace is_customer_066 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw466' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_067;
create regular tablespace is_customer_067 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw467' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_068;
create regular tablespace is_customer_068 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw468' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
```

```
commit;
drop tablespace is_customer_069;
create regular tablespace is_customer_069 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw469' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_070;
create regular tablespace is_customer_070 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw470' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_071;
create regular tablespace is_customer_071 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw471' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_072;
create regular tablespace is_customer_072 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw472' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_073;
create regular tablespace is_customer_073 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw473' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_074;
create regular tablespace is_customer_074 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw474' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_075;
create regular tablespace is_customer_075 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw475' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_076;
create regular tablespace is_customer_076 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw476' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_077;
create regular tablespace is_customer_077 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw477' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_078;
create regular tablespace is_customer_078 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw478' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_079;
create regular tablespace is_customer_079 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw479' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_customer_080;
create regular tablespace is_customer_080 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw480' 104448
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;




drop tablespace ts_history_001;
create regular tablespace ts_history_001 pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw481' 79616
             )
       extentsize 32
       bufferpool IBMDEFAULT16K
       prefetchsize 4096;
commit;
drop tablespace ts_history_002;
create regular tablespace ts_history_002 pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw482' 79616
             )
       extentsize 32
       bufferpool IBMDEFAULT16K
       prefetchsize 4096;
commit;
drop tablespace ts_history_003;
create regular tablespace ts_history_003 pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw483' 79616
             )
       extentsize 32
       bufferpool IBMDEFAULT16K
       prefetchsize 4096;
commit;
drop tablespace ts_history_004;
create regular tablespace ts_history_004 pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw484' 79616
             )
       extentsize 32
       bufferpool IBMDEFAULT16K
       prefetchsize 4096;
commit;
drop tablespace ts_history_005;
create regular tablespace ts_history_005 pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw485' 79616
             )
       extentsize 32
       bufferpool IBMDEFAULT16K
       prefetchsize 4096;
commit;
drop tablespace ts_history_006;
create regular tablespace ts_history_006 pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw486' 79616
             )
       extentsize 32
       bufferpool IBMDEFAULT16K
       prefetchsize 4096;
commit;
drop tablespace ts_history_007;
create regular tablespace ts_history_007 pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw487' 79616
```

```
            )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_008;
create regular tablespace ts_history_008 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw488' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_009;
create regular tablespace ts_history_009 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw489' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_010;
create regular tablespace ts_history_010 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw490' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_011;
create regular tablespace ts_history_011 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw491' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_012;
create regular tablespace ts_history_012 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw492' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_013;
create regular tablespace ts_history_013 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw493' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_014;
create regular tablespace ts_history_014 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw494' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_015;
create regular tablespace ts_history_015 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw495' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_016;
create regular tablespace ts_history_016 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw496' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_017;
create regular tablespace ts_history_017 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw497' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_018;
create regular tablespace ts_history_018 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw498' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_019;
create regular tablespace ts_history_019 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw499' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_020;
create regular tablespace ts_history_020 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw500' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_021;
create regular tablespace ts_history_021 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw501' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;

drop tablespace ts_history_022;
create regular tablespace ts_history_022 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw502' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_023;
create regular tablespace ts_history_023 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw503' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_024;
create regular tablespace ts_history_024 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw504' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_025;
create regular tablespace ts_history_025 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw505' 79616
               )
        extentsize 32
        bufferpool IBMDEFAULT16K
        prefetchsize 4096;
commit;
drop tablespace ts_history_026;
create regular tablespace ts_history_026 pagesize 16K
         managed by database
         using
        (
                device '/dev/raw/raw506' 79616
```

```
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_027;
create regular tablespace ts_history_027 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw507' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_028;
create regular tablespace ts_history_028 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw508' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_029;
create regular tablespace ts_history_029 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw509' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_030;
create regular tablespace ts_history_030 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw510' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_031;
create regular tablespace ts_history_031 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw511' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_032;
create regular tablespace ts_history_032 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw512' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_033;
create regular tablespace ts_history_033 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw513' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_034;
create regular tablespace ts_history_034 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw514' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_035;
create regular tablespace ts_history_035 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw515' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_036;
create regular tablespace ts_history_036 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw516' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_037;
create regular tablespace ts_history_037 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw517' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_038;
create regular tablespace ts_history_038 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw518' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_039;
create regular tablespace ts_history_039 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw519' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_040;
create regular tablespace ts_history_040 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw520' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;


drop tablespace ts_history_041;
create regular tablespace ts_history_041 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw521' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_042;
create regular tablespace ts_history_042 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw522' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_043;
create regular tablespace ts_history_043 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw523' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_044;
create regular tablespace ts_history_044 pagesize 16K
     managed by database
     using
    (
          device '/dev/raw/raw524' 79616
        )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_045;
create regular tablespace ts_history_045 pagesize 16K
     managed by database
     using
    (
```

```
            device '/dev/raw/raw525' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_046;
create regular tablespace ts_history_046 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw526' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_047;
create regular tablespace ts_history_047 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw527' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_048;
create regular tablespace ts_history_048 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw528' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_049;
create regular tablespace ts_history_049 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw529' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_050;
create regular tablespace ts_history_050 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw530' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_051;
create regular tablespace ts_history_051 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw531' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_052;
create regular tablespace ts_history_052 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw532' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_053;
create regular tablespace ts_history_053 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw533' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_054;
create regular tablespace ts_history_054 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw534' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_055;
create regular tablespace ts_history_055 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw535' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_056;
create regular tablespace ts_history_056 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw536' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_057;
create regular tablespace ts_history_057 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw537' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_058;
create regular tablespace ts_history_058 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw538' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_059;
create regular tablespace ts_history_059 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw539' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_060;
create regular tablespace ts_history_060 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw540' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_061;
create regular tablespace ts_history_061 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw541' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;

drop tablespace ts_history_062;
create regular tablespace ts_history_062 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw542' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_063;
create regular tablespace ts_history_063 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw543' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_064;
create regular tablespace ts_history_064 pagesize 16K
       managed by database
       using
      (
```

```
            device '/dev/raw/raw544' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_065;
create regular tablespace ts_history_065 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw545' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_066;
create regular tablespace ts_history_066 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw546' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_067;
create regular tablespace ts_history_067 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw547' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_068;
create regular tablespace ts_history_068 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw548' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_069;
create regular tablespace ts_history_069 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw549' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_070;
create regular tablespace ts_history_070 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw550' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_071;
create regular tablespace ts_history_071 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw551' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_072;
create regular tablespace ts_history_072 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw552' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_073;
create regular tablespace ts_history_073 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw553' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_074;
create regular tablespace ts_history_074 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw554' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_075;
create regular tablespace ts_history_075 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw555' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_076;
create regular tablespace ts_history_076 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw556' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_077;
create regular tablespace ts_history_077 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw557' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_078;
create regular tablespace ts_history_078 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw558' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_079;
create regular tablespace ts_history_079 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw559' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;
drop tablespace ts_history_080;
create regular tablespace ts_history_080 pagesize 16K
       managed by database
       using
      (
            device '/dev/raw/raw560' 79616
           )
      extentsize 32
      bufferpool IBMDEFAULT16K
      prefetchsize 4096;
commit;



drop tablespace ts_order_001;
create regular tablespace ts_order_001 pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw561' 100864
           )
      extentsize 32
      bufferpool IBMDEFAULT8K
      prefetchsize 4096;
commit;
drop tablespace ts_order_002;
create regular tablespace ts_order_002 pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw562' 100864
           )
      extentsize 32
      bufferpool IBMDEFAULT8K
      prefetchsize 4096;
commit;
drop tablespace ts_order_003;
create regular tablespace ts_order_003 pagesize 8K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw563' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_004;
create regular tablespace ts_order_004 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw564' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_005;
create regular tablespace ts_order_005 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw565' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_006;
create regular tablespace ts_order_006 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw566' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_007;
create regular tablespace ts_order_007 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw567' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_008;
create regular tablespace ts_order_008 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw568' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_009;
create regular tablespace ts_order_009 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw569' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_010;
create regular tablespace ts_order_010 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw570' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_011;
create regular tablespace ts_order_011 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw571' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_012;
create regular tablespace ts_order_012 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw572' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_013;
create regular tablespace ts_order_013 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw573' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_014;
create regular tablespace ts_order_014 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw574' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_015;
create regular tablespace ts_order_015 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw575' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_016;
create regular tablespace ts_order_016 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw576' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_017;
create regular tablespace ts_order_017 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw577' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_018;
create regular tablespace ts_order_018 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw578' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_019;
create regular tablespace ts_order_019 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw579' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_020;
create regular tablespace ts_order_020 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw580' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_021;
create regular tablespace ts_order_021 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw581' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;

drop tablespace ts_order_022;
create regular tablespace ts_order_022 pagesize 8K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw582' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_023;
create regular tablespace ts_order_023 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw583' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_024;
create regular tablespace ts_order_024 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw584' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_025;
create regular tablespace ts_order_025 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw585' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_026;
create regular tablespace ts_order_026 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw586' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_027;
create regular tablespace ts_order_027 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw587' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_028;
create regular tablespace ts_order_028 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw588' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_029;
create regular tablespace ts_order_029 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw589' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_030;
create regular tablespace ts_order_030 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw590' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_031;
create regular tablespace ts_order_031 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw591' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_032;
create regular tablespace ts_order_032 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw592' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_033;
create regular tablespace ts_order_033 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw593' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_034;
create regular tablespace ts_order_034 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw594' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_035;
create regular tablespace ts_order_035 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw595' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_036;
create regular tablespace ts_order_036 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw596' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_037;
create regular tablespace ts_order_037 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw597' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_038;
create regular tablespace ts_order_038 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw598' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_039;
create regular tablespace ts_order_039 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw599' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_040;
create regular tablespace ts_order_040 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw600' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;


drop tablespace ts_order_041;
```

```
create regular tablespace ts_order_041 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw601' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_042;
create regular tablespace ts_order_042 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw602' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_043;
create regular tablespace ts_order_043 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw603' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_044;
create regular tablespace ts_order_044 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw604' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_045;
create regular tablespace ts_order_045 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw605' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_046;
create regular tablespace ts_order_046 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw606' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_047;
create regular tablespace ts_order_047 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw607' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_048;
create regular tablespace ts_order_048 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw608' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_049;
create regular tablespace ts_order_049 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw609' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_050;
create regular tablespace ts_order_050 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw610' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_051;
create regular tablespace ts_order_051 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw611' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_052;
create regular tablespace ts_order_052 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw612' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_053;
create regular tablespace ts_order_053 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw613' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_054;
create regular tablespace ts_order_054 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw614' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_055;
create regular tablespace ts_order_055 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw615' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_056;
create regular tablespace ts_order_056 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw616' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_057;
create regular tablespace ts_order_057 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw617' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_058;
create regular tablespace ts_order_058 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw618' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_059;
create regular tablespace ts_order_059 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw619' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_060;
create regular tablespace ts_order_060 pagesize 8K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw620' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_061;
create regular tablespace ts_order_061 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw621' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;

drop tablespace ts_order_062;
create regular tablespace ts_order_062 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw622' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_063;
create regular tablespace ts_order_063 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw623' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_064;
create regular tablespace ts_order_064 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw624' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_065;
create regular tablespace ts_order_065 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw625' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_066;
create regular tablespace ts_order_066 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw626' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_067;
create regular tablespace ts_order_067 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw627' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_068;
create regular tablespace ts_order_068 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw628' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_069;
create regular tablespace ts_order_069 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw629' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_070;
create regular tablespace ts_order_070 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw630' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_071;
create regular tablespace ts_order_071 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw631' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_072;
create regular tablespace ts_order_072 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw632' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_073;
create regular tablespace ts_order_073 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw633' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_074;
create regular tablespace ts_order_074 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw634' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_075;
create regular tablespace ts_order_075 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw635' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_076;
create regular tablespace ts_order_076 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw636' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_077;
create regular tablespace ts_order_077 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw637' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_078;
create regular tablespace ts_order_078 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw638' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_079;
create regular tablespace ts_order_079 pagesize 8K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw639' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_order_080;
create regular tablespace ts_order_080 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw640' 100864
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;



drop tablespace is_order_001;
create regular tablespace is_order_001 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw641' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_002;
create regular tablespace is_order_002 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw642' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_003;
create regular tablespace is_order_003 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw643' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_004;
create regular tablespace is_order_004 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw644' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_005;
create regular tablespace is_order_005 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw645' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_006;
create regular tablespace is_order_006 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw646' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_007;
create regular tablespace is_order_007 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw647' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_008;
create regular tablespace is_order_008 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw648' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_009;
create regular tablespace is_order_009 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw649' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_010;
create regular tablespace is_order_010 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw650' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_011;
create regular tablespace is_order_011 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw651' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_012;
create regular tablespace is_order_012 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw652' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_013;
create regular tablespace is_order_013 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw653' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_014;
create regular tablespace is_order_014 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw654' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_015;
create regular tablespace is_order_015 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw655' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_016;
create regular tablespace is_order_016 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw656' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_017;
create regular tablespace is_order_017 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw657' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
```

```
commit;
drop tablespace is_order_018;
create regular tablespace is_order_018 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw658' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_019;
create regular tablespace is_order_019 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw659' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_020;
create regular tablespace is_order_020 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw660' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_021;
create regular tablespace is_order_021 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw661' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;

drop tablespace is_order_022;
create regular tablespace is_order_022 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw662' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_023;
create regular tablespace is_order_023 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw663' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_024;
create regular tablespace is_order_024 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw664' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_025;
create regular tablespace is_order_025 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw665' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_026;
create regular tablespace is_order_026 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw666' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_027;
create regular tablespace is_order_027 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw667' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_028;
create regular tablespace is_order_028 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw668' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_029;
create regular tablespace is_order_029 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw669' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_030;
create regular tablespace is_order_030 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw670' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_031;
create regular tablespace is_order_031 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw671' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_032;
create regular tablespace is_order_032 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw672' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_033;
create regular tablespace is_order_033 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw673' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_034;
create regular tablespace is_order_034 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw674' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_035;
create regular tablespace is_order_035 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw675' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_036;
create regular tablespace is_order_036 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw676' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
```

```
commit;
drop tablespace is_order_037;
create regular tablespace is_order_037 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw677' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_038;
create regular tablespace is_order_038 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw678' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_039;
create regular tablespace is_order_039 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw679' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_040;
create regular tablespace is_order_040 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw680' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;


drop tablespace is_order_041;
create regular tablespace is_order_041 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw681' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_042;
create regular tablespace is_order_042 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw682' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_043;
create regular tablespace is_order_043 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw683' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_044;
create regular tablespace is_order_044 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw684' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_045;
create regular tablespace is_order_045 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw685' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_046;
create regular tablespace is_order_046 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw686' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_047;
create regular tablespace is_order_047 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw687' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_048;
create regular tablespace is_order_048 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw688' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_049;
create regular tablespace is_order_049 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw689' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_050;
create regular tablespace is_order_050 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw690' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_051;
create regular tablespace is_order_051 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw691' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_052;
create regular tablespace is_order_052 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw692' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_053;
create regular tablespace is_order_053 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw693' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_054;
create regular tablespace is_order_054 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw694' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_055;
create regular tablespace is_order_055 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw695' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
```

```
        prefetchsize 4096;
commit;
drop tablespace is_order_056;
create regular tablespace is_order_056 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw696' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_057;
create regular tablespace is_order_057 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw697' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_058;
create regular tablespace is_order_058 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw698' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_059;
create regular tablespace is_order_059 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw699' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_060;
create regular tablespace is_order_060 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw700' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_061;
create regular tablespace is_order_061 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw701' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;

drop tablespace is_order_062;
create regular tablespace is_order_062 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw702' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_063;
create regular tablespace is_order_063 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw703' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_064;
create regular tablespace is_order_064 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw704' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_065;
create regular tablespace is_order_065 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw705' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_066;
create regular tablespace is_order_066 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw706' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_067;
create regular tablespace is_order_067 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw707' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_068;
create regular tablespace is_order_068 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw708' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_069;
create regular tablespace is_order_069 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw709' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_070;
create regular tablespace is_order_070 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw710' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_071;
create regular tablespace is_order_071 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw711' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_072;
create regular tablespace is_order_072 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw712' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_073;
create regular tablespace is_order_073 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw713' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace is_order_074;
create regular tablespace is_order_074 pagesize 8K
        managed by database
        using
        (
                device '/dev/raw/raw714' 86272
                )
        extentsize 32
        bufferpool IBMDEFAULT8K
```

```
        prefetchsize 4096;
commit;
drop tablespace is_order_075;
create regular tablespace is_order_075 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw715' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_076;
create regular tablespace is_order_076 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw716' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_077;
create regular tablespace is_order_077 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw717' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_078;
create regular tablespace is_order_078 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw718' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_079;
create regular tablespace is_order_079 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw719' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace is_order_080;
create regular tablespace is_order_080 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw720' 86272
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
```

```
drop tablespace ts_orderline_001;
create regular tablespace ts_orderline_001 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw721' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_002;
create regular tablespace ts_orderline_002 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw722' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_003;
create regular tablespace ts_orderline_003 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw723' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_004;
create regular tablespace ts_orderline_004 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw724' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_005;
create regular tablespace ts_orderline_005 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw725' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_006;
create regular tablespace ts_orderline_006 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw726' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_007;
create regular tablespace ts_orderline_007 pagesize 8K
        managed by database
        using
```

```
       (
              device '/dev/raw/raw727' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_008;
create regular tablespace ts_orderline_008 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw728' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_009;
create regular tablespace ts_orderline_009 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw729' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_010;
create regular tablespace ts_orderline_010 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw730' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_011;
create regular tablespace ts_orderline_011 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw731' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_012;
create regular tablespace ts_orderline_012 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw732' 2610176
             )
       extentsize 32
       bufferpool IBMDEFAULT8K
       prefetchsize 4096;
commit;
drop tablespace ts_orderline_013;
create regular tablespace ts_orderline_013 pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw733' 2610176
             )
       extentsize 32
```

```
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_014;
create regular tablespace ts_orderline_014 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw734' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_015;
create regular tablespace ts_orderline_015 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw735' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_016;
create regular tablespace ts_orderline_016 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw736' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_017;
create regular tablespace ts_orderline_017 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw737' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_018;
create regular tablespace ts_orderline_018 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw738' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_019;
create regular tablespace ts_orderline_019 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw739' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_020;
create regular tablespace ts_orderline_020 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw740' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_021;
create regular tablespace ts_orderline_021 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw741' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;

drop tablespace ts_orderline_022;
create regular tablespace ts_orderline_022 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw742' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_023;
create regular tablespace ts_orderline_023 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw743' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_024;
create regular tablespace ts_orderline_024 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw744' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_025;
create regular tablespace ts_orderline_025 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw745' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_026;
create regular tablespace ts_orderline_026 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw746' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_027;
create regular tablespace ts_orderline_027 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw747' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_028;
create regular tablespace ts_orderline_028 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw748' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_029;
create regular tablespace ts_orderline_029 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw749' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_030;
create regular tablespace ts_orderline_030 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw750' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_031;
create regular tablespace ts_orderline_031 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw751' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_032;
create regular tablespace ts_orderline_032 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw752' 2610176
               )
        extentsize 32
```

```
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_033;
create regular tablespace ts_orderline_033 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw753' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_034;
create regular tablespace ts_orderline_034 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw754' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_035;
create regular tablespace ts_orderline_035 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw755' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_036;
create regular tablespace ts_orderline_036 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw756' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_037;
create regular tablespace ts_orderline_037 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw757' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_038;
create regular tablespace ts_orderline_038 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw758' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_039;
create regular tablespace ts_orderline_039 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw759' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_040;
create regular tablespace ts_orderline_040 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw760' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;


drop tablespace ts_orderline_041;
create regular tablespace ts_orderline_041 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw761' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_042;
create regular tablespace ts_orderline_042 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw762' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_043;
create regular tablespace ts_orderline_043 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw763' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_044;
create regular tablespace ts_orderline_044 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw764' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_045;
create regular tablespace ts_orderline_045 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw765' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_046;
create regular tablespace ts_orderline_046 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw766' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_047;
create regular tablespace ts_orderline_047 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw767' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_048;
create regular tablespace ts_orderline_048 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw768' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_049;
create regular tablespace ts_orderline_049 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw769' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_050;
create regular tablespace ts_orderline_050 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw770' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_051;
create regular tablespace ts_orderline_051 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw771' 2610176
               )
```

```
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_052;
create regular tablespace ts_orderline_052 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw772' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_053;
create regular tablespace ts_orderline_053 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw773' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_054;
create regular tablespace ts_orderline_054 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw774' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_055;
create regular tablespace ts_orderline_055 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw775' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_056;
create regular tablespace ts_orderline_056 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw776' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_057;
create regular tablespace ts_orderline_057 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw777' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_058;
create regular tablespace ts_orderline_058 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw778' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_059;
create regular tablespace ts_orderline_059 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw779' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_060;
create regular tablespace ts_orderline_060 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw780' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_061;
create regular tablespace ts_orderline_061 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw781' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;

drop tablespace ts_orderline_062;
create regular tablespace ts_orderline_062 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw782' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_063;
create regular tablespace ts_orderline_063 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw783' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_064;
create regular tablespace ts_orderline_064 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw784' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_065;
create regular tablespace ts_orderline_065 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw785' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_066;
create regular tablespace ts_orderline_066 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw786' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_067;
create regular tablespace ts_orderline_067 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw787' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_068;
create regular tablespace ts_orderline_068 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw788' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_069;
create regular tablespace ts_orderline_069 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw789' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_070;
create regular tablespace ts_orderline_070 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw790' 2610176
               )
```

```
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_071;
create regular tablespace ts_orderline_071 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw791' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_072;
create regular tablespace ts_orderline_072 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw792' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_073;
create regular tablespace ts_orderline_073 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw793' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_074;
create regular tablespace ts_orderline_074 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw794' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_075;
create regular tablespace ts_orderline_075 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw795' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_076;
create regular tablespace ts_orderline_076 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw796' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
```

```
drop tablespace ts_orderline_077;
create regular tablespace ts_orderline_077 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw797' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_078;
create regular tablespace ts_orderline_078 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw798' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_079;
create regular tablespace ts_orderline_079 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw799' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
drop tablespace ts_orderline_080;
create regular tablespace ts_orderline_080 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw800' 2610176
               )
        extentsize 32
        bufferpool IBMDEFAULT8K
        prefetchsize 4096;
commit;
```

```
drop tablespace ts_newordA_001;
create regular tablespace ts_newordA_001 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw801' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_newordA_002;
create regular tablespace ts_newordA_002 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw802' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
```

```
commit;
drop tablespace ts_newordA_003;
create regular tablespace ts_newordA_003 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw803' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_newordA_004;
create regular tablespace ts_newordA_004 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw804' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_newordA_005;
create regular tablespace ts_newordA_005 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw805' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_newordA_006;
create regular tablespace ts_newordA_006 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw806' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_newordA_007;
create regular tablespace ts_newordA_007 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw807' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_newordA_008;
create regular tablespace ts_newordA_008 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw808' 50432
               )
        extentsize 32
        bufferpool IBMDEFAULTBP
        prefetchsize 4096;
commit;
drop tablespace ts_newordA_009;
create regular tablespace ts_newordA_009 pagesize 4K
         managed by database
```

```
        using
       (
              device '/dev/raw/raw809' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_010;
create regular tablespace ts_newordA_010 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw810' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_011;
create regular tablespace ts_newordA_011 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw811' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_012;
create regular tablespace ts_newordA_012 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw812' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_013;
create regular tablespace ts_newordA_013 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw813' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_014;
create regular tablespace ts_newordA_014 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw814' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_015;
create regular tablespace ts_newordA_015 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw815' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_016;
create regular tablespace ts_newordA_016 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw816' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_017;
create regular tablespace ts_newordA_017 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw817' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_018;
create regular tablespace ts_newordA_018 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw818' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_019;
create regular tablespace ts_newordA_019 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw819' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_020;
create regular tablespace ts_newordA_020 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw820' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_021;
create regular tablespace ts_newordA_021 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw821' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_022;
create regular tablespace ts_newordA_022 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw822' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_023;
create regular tablespace ts_newordA_023 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw823' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_024;
create regular tablespace ts_newordA_024 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw824' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_025;
create regular tablespace ts_newordA_025 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw825' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_026;
create regular tablespace ts_newordA_026 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw826' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_027;
create regular tablespace ts_newordA_027 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw827' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_028;
create regular tablespace ts_newordA_028 pagesize 4K
        managed by database
```

```
        using
       (
              device '/dev/raw/raw828' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_029;
create regular tablespace ts_newordA_029 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw829' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_030;
create regular tablespace ts_newordA_030 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw830' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_031;
create regular tablespace ts_newordA_031 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw831' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_032;
create regular tablespace ts_newordA_032 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw832' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_033;
create regular tablespace ts_newordA_033 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw833' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_034;
create regular tablespace ts_newordA_034 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw834' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_035;
create regular tablespace ts_newordA_035 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw835' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_036;
create regular tablespace ts_newordA_036 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw836' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_037;
create regular tablespace ts_newordA_037 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw837' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_038;
create regular tablespace ts_newordA_038 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw838' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_039;
create regular tablespace ts_newordA_039 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw839' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_040;
create regular tablespace ts_newordA_040 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw840' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_041;
create regular tablespace ts_newordA_041 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw841' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_042;
create regular tablespace ts_newordA_042 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw842' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_043;
create regular tablespace ts_newordA_043 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw843' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_044;
create regular tablespace ts_newordA_044 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw844' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_045;
create regular tablespace ts_newordA_045 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw845' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_046;
create regular tablespace ts_newordA_046 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw846' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_047;
create regular tablespace ts_newordA_047 pagesize 4K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw847' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_048;
create regular tablespace ts_newordA_048 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw848' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_049;
create regular tablespace ts_newordA_049 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw849' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_050;
create regular tablespace ts_newordA_050 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw850' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_051;
create regular tablespace ts_newordA_051 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw851' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_052;
create regular tablespace ts_newordA_052 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw852' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_053;
create regular tablespace ts_newordA_053 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw853' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_054;
create regular tablespace ts_newordA_054 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw854' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_055;
create regular tablespace ts_newordA_055 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw855' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_056;
create regular tablespace ts_newordA_056 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw856' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_057;
create regular tablespace ts_newordA_057 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw857' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_058;
create regular tablespace ts_newordA_058 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw858' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_059;
create regular tablespace ts_newordA_059 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw859' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_060;
create regular tablespace ts_newordA_060 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw860' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_061;
create regular tablespace ts_newordA_061 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw861' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;

drop tablespace ts_newordA_062;
create regular tablespace ts_newordA_062 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw862' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_063;
create regular tablespace ts_newordA_063 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw863' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_064;
create regular tablespace ts_newordA_064 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw864' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_065;
create regular tablespace ts_newordA_065 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw865' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_066;
create regular tablespace ts_newordA_066 pagesize 4K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw866' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_067;
create regular tablespace ts_newordA_067 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw867' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_068;
create regular tablespace ts_newordA_068 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw868' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_069;
create regular tablespace ts_newordA_069 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw869' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_070;
create regular tablespace ts_newordA_070 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw870' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_071;
create regular tablespace ts_newordA_071 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw871' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_072;
create regular tablespace ts_newordA_072 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw872' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_073;
create regular tablespace ts_newordA_073 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw873' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_074;
create regular tablespace ts_newordA_074 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw874' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_075;
create regular tablespace ts_newordA_075 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw875' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_076;
create regular tablespace ts_newordA_076 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw876' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_077;
create regular tablespace ts_newordA_077 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw877' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_078;
create regular tablespace ts_newordA_078 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw878' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_079;
create regular tablespace ts_newordA_079 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw879' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordA_080;
create regular tablespace ts_newordA_080 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw880' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;

drop tablespace ts_newordB_001;
create regular tablespace ts_newordB_001 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw881' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_002;
create regular tablespace ts_newordB_002 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw882' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_003;
create regular tablespace ts_newordB_003 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw883' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_004;
create regular tablespace ts_newordB_004 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw884' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_005;
create regular tablespace ts_newordB_005 pagesize 4K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw885' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_006;
create regular tablespace ts_newordB_006 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw886' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_007;
create regular tablespace ts_newordB_007 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw887' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_008;
create regular tablespace ts_newordB_008 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw888' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_009;
create regular tablespace ts_newordB_009 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw889' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_010;
create regular tablespace ts_newordB_010 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw890' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_011;
create regular tablespace ts_newordB_011 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw891' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_012;
create regular tablespace ts_newordB_012 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw892' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_013;
create regular tablespace ts_newordB_013 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw893' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_014;
create regular tablespace ts_newordB_014 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw894' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_015;
create regular tablespace ts_newordB_015 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw895' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_016;
create regular tablespace ts_newordB_016 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw896' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_017;
create regular tablespace ts_newordB_017 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw897' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_018;
create regular tablespace ts_newordB_018 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw898' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_019;
create regular tablespace ts_newordB_019 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw899' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_020;
create regular tablespace ts_newordB_020 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw900' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_021;
create regular tablespace ts_newordB_021 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw901' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;

drop tablespace ts_newordB_022;
create regular tablespace ts_newordB_022 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw902' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_023;
create regular tablespace ts_newordB_023 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw903' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_024;
create regular tablespace ts_newordB_024 pagesize 4K
```

```
        managed by database
        using
       (
              device '/dev/raw/raw904' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_025;
create regular tablespace ts_newordB_025 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw905' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_026;
create regular tablespace ts_newordB_026 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw906' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_027;
create regular tablespace ts_newordB_027 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw907' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_028;
create regular tablespace ts_newordB_028 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw908' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_029;
create regular tablespace ts_newordB_029 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw909' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_030;
create regular tablespace ts_newordB_030 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw910' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_031;
create regular tablespace ts_newordB_031 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw911' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_032;
create regular tablespace ts_newordB_032 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw912' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_033;
create regular tablespace ts_newordB_033 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw913' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_034;
create regular tablespace ts_newordB_034 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw914' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_035;
create regular tablespace ts_newordB_035 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw915' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_036;
create regular tablespace ts_newordB_036 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw916' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_037;
create regular tablespace ts_newordB_037 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw917' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_038;
create regular tablespace ts_newordB_038 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw918' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_039;
create regular tablespace ts_newordB_039 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw919' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_040;
create regular tablespace ts_newordB_040 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw920' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;


drop tablespace ts_newordB_041;
create regular tablespace ts_newordB_041 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw921' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_042;
create regular tablespace ts_newordB_042 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw922' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_043;
```

```
create regular tablespace ts_newordB_043 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw923' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_044;
create regular tablespace ts_newordB_044 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw924' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_045;
create regular tablespace ts_newordB_045 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw925' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_046;
create regular tablespace ts_newordB_046 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw926' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_047;
create regular tablespace ts_newordB_047 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw927' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_048;
create regular tablespace ts_newordB_048 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw928' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_049;
create regular tablespace ts_newordB_049 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw929' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_050;
create regular tablespace ts_newordB_050 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw930' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_051;
create regular tablespace ts_newordB_051 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw931' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_052;
create regular tablespace ts_newordB_052 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw932' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_053;
create regular tablespace ts_newordB_053 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw933' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_054;
create regular tablespace ts_newordB_054 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw934' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_055;
create regular tablespace ts_newordB_055 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw935' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_056;
create regular tablespace ts_newordB_056 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw936' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_057;
create regular tablespace ts_newordB_057 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw937' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_058;
create regular tablespace ts_newordB_058 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw938' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_059;
create regular tablespace ts_newordB_059 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw939' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_060;
create regular tablespace ts_newordB_060 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw940' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_061;
create regular tablespace ts_newordB_061 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw941' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;

drop tablespace ts_newordB_062;
```

```
create regular tablespace ts_newordB_062 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw942' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_063;
create regular tablespace ts_newordB_063 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw943' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_064;
create regular tablespace ts_newordB_064 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw944' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_065;
create regular tablespace ts_newordB_065 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw945' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_066;
create regular tablespace ts_newordB_066 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw946' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_067;
create regular tablespace ts_newordB_067 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw947' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_068;
create regular tablespace ts_newordB_068 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw948' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_069;
create regular tablespace ts_newordB_069 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw949' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_070;
create regular tablespace ts_newordB_070 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw950' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_071;
create regular tablespace ts_newordB_071 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw951' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_072;
create regular tablespace ts_newordB_072 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw952' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_073;
create regular tablespace ts_newordB_073 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw953' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_074;
create regular tablespace ts_newordB_074 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw954' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_075;
create regular tablespace ts_newordB_075 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw955' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_076;
create regular tablespace ts_newordB_076 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw956' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_077;
create regular tablespace ts_newordB_077 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw957' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_078;
create regular tablespace ts_newordB_078 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw958' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_079;
create regular tablespace ts_newordB_079 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw959' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;
drop tablespace ts_newordB_080;
create regular tablespace ts_newordB_080 pagesize 4K
        managed by database
        using
       (
              device '/dev/raw/raw960' 50432
             )
       extentsize 32
       bufferpool IBMDEFAULTBP
       prefetchsize 4096;
commit;

connect reset;
```

bp/alter_bufferpool.ddl

```
connect to tpcc;

alter bufferpool IBMDEFAULTBP size 200;

alter bufferpool ITM        size 2526;

alter bufferpool WDS1       size 1009;
alter bufferpool WDS2       size 1009;
alter bufferpool WDS3       size 1009;
alter bufferpool WDS4       size 1009;
alter bufferpool WDS5       size 1009;
alter bufferpool WDS6       size 1009;
alter bufferpool WDS7       size 1009;
alter bufferpool WDS8       size 1009;
alter bufferpool WDS9       size 1009;
alter bufferpool WDS10      size 1009;
alter bufferpool WDS11      size 1009;
alter bufferpool WDS12      size 1009;
alter bufferpool WDS13      size 1009;
alter bufferpool WDS14      size 1009;
alter bufferpool WDS15      size 1009;
alter bufferpool WDS16      size 1009;


alter bufferpool CST1       size 1008;
alter bufferpool CST2       size 1008;
alter bufferpool CST3       size 1008;
alter bufferpool CST4       size 1008;
alter bufferpool CST5       size 1008;
alter bufferpool CST6       size 1008;
alter bufferpool CST7       size 1008;
alter bufferpool CST8       size 1008;
alter bufferpool CST9       size 1008;
alter bufferpool CST10      size 1008;
alter bufferpool CST11      size 1008;
alter bufferpool CST12      size 1008;
alter bufferpool CST13      size 1008;
alter bufferpool CST14      size 1008;
alter bufferpool CST15      size 1008;
alter bufferpool CST16      size 1008;

alter bufferpool NEW1       size 50080;
alter bufferpool NEW2       size 50080;
alter bufferpool NEW3       size 50080;
alter bufferpool NEW4       size 50080;
alter bufferpool NEW5       size 50080;
alter bufferpool NEW6       size 50080;
alter bufferpool NEW7       size 50080;
alter bufferpool NEW8       size 50080;
alter bufferpool NEW9       size 50080;
alter bufferpool NEW10      size 50080;
alter bufferpool NEW11      size 50080;
alter bufferpool NEW12      size 50080;
alter bufferpool NEW13      size 50080;
alter bufferpool NEW14      size 50080;
alter bufferpool NEW15      size 50080;
alter bufferpool NEW16      size 50080;

alter bufferpool OLNORDIORD1 size 290000;
alter bufferpool OLNORDIORD2 size 290000;
alter bufferpool OLNORDIORD3 size 290000;
alter bufferpool OLNORDIORD4 size 290000;
alter bufferpool OLNORDIORD5 size 290000;
alter bufferpool OLNORDIORD6 size 290000;
alter bufferpool OLNORDIORD7 size 290000;
alter bufferpool OLNORDIORD8 size 290000;
alter bufferpool OLNORDIORD9 size 290000;
alter bufferpool OLNORDIORD10 size 290000;
alter bufferpool OLNORDIORD11 size 290000;
alter bufferpool OLNORDIORD12 size 290000;
alter bufferpool OLNORDIORD13 size 290000;
alter bufferpool OLNORDIORD14 size 290000;
alter bufferpool OLNORDIORD15 size 290000;
alter bufferpool OLNORDIORD16 size 290000;

alter bufferpool HST1       size 1004;
alter bufferpool HST2       size 1004;
alter bufferpool HST3       size 1004;
alter bufferpool HST4       size 1004;
alter bufferpool HST5       size 1004;
alter bufferpool HST6       size 1004;
alter bufferpool HST7       size 1004;
alter bufferpool HST8       size 1004;
alter bufferpool HST9       size 1004;
alter bufferpool HST10      size 1004;
alter bufferpool HST11      size 1004;
alter bufferpool HST12      size 1004;
alter bufferpool HST13      size 1004;
alter bufferpool HST14      size 1004;
alter bufferpool HST15      size 1004;
alter bufferpool HST16      size 1004;

alter bufferpool CSTI1      size 39600;
alter bufferpool CSTI2      size 39600;
alter bufferpool CSTI3      size 39600;
alter bufferpool CSTI4      size 39600;
alter bufferpool CSTI5      size 39600;
alter bufferpool CSTI6      size 39600;
alter bufferpool CSTI7      size 39600;
alter bufferpool CSTI8      size 39600;
alter bufferpool CSTI9      size 39600;
alter bufferpool CSTI10     size 39600;
alter bufferpool CSTI11     size 39600;
alter bufferpool CSTI12     size 39600;
alter bufferpool CSTI13     size 39600;
alter bufferpool CSTI14     size 39600;
alter bufferpool CSTI15     size 39600;
alter bufferpool CSTI16     size 39600;

alter bufferpool STK1       size 1124992;
alter bufferpool STK2       size 1124992;
alter bufferpool STK3       size 1124992;
alter bufferpool STK4       size 1124992;
alter bufferpool STK5       size 1124992;
alter bufferpool STK6       size 1124992;
alter bufferpool STK7       size 1124992;
alter bufferpool STK8       size 1124992;
alter bufferpool STK9       size 1124992;
alter bufferpool STK10      size 1124992;
alter bufferpool STK11      size 1124992;
alter bufferpool STK12      size 1124992;
alter bufferpool STK13      size 1124992;
alter bufferpool STK14      size 1124992;
alter bufferpool STK15      size 1124992;
alter bufferpool STK16      size 1124992;

connect reset;
terminate;
```

11.2. Data Generation

Makefile.config

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
######

#
# Makefile.config - Linux 64-bit
#
#



# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g", "-DDEBUGIT" "-g -DDEBUGIT" or left blank
CC=cc
CFLAGS_OS=-DSQLUNIX -DSQLLinux -O2 -fpic -m64
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=gcc
LD_STORP=gcc
LDFLAGS_EXEC=
LDFLAGS_SHLIB=-shared
LDFLAGS_STORP=$(LDFLAGS_SHLIB)
LDFLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2 -m64
LDFLAGS_OUT=-o

# Library Configuration
AR=ar
ARFLAGS=-rv
ARFLAGS_LIB=
ARFLAGS_OUT=

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

# OS File Extensions & Path Separators
OBJEXT=.o
LIBEXT=.a
SHLIBEXT=.so
BINEXT=
SLASH=/
CMDSEP=;
```

Src.Common/Makefile

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
```

```
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
######

#
# Makefile - Makefile for Src.Common
#
#

include $(TPCC_ROOT)/Makefile.config

#
################################################################################
# Preprocessor, Compiler and LInker Flags
#
################################################################################

BND_OPTS =    GRANT PUBLIC \
        MESSAGES $*.bnd.msg
PRP_OPTS =    BINDFILE \
        OPTLEVEL 1 \
        ISOLATION RR \
        MESSAGES $*.prep.msg \
        LEVEL $(TPCC_VERSION) \
        NOLINEMACRO

INCLUDE =   -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =    $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDE) \
        -DSQLA_NOLINES -D$(DB2EDITION) -D$(DB2VERSION) \
        -D$(TPCC_SPTYPE)

UTIL_OBJ_DBG =tpccdbg$(OBJEXT)
UTIL_OBJ_GEN =tpccmisc$(OBJEXT)
UTIL_OBJ_DB2 = tpccctx$(OBJEXT)

#
################################################################################
# User Targets
#
################################################################################

all:    $(UTIL_OBJ_DBG) $(UTIL_OBJ_GEN) connect $(UTIL_OBJ_DB2) disconnect

dbgen:  $(UTIL_OBJ_GEN)

clean:
    - $(ERASE) *$(OBJEXT) *.bnd *.msg tpccctx.c

#
################################################################################
# Helper Targets
#
################################################################################

connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

rebind:  connect
    db2 bind tpccctx.bnd $(BND_OPTS)
```

```
#
################################################################################
# Build Rules
#
################################################################################

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
    @echo "Prepping $*.sqc"
    -db2 prep $*.sqc $(PRP_OPTS)
    @echo "Binding $*.bnd"
    db2 bind $*.bnd $(BND_OPTS)

#
################################################################################
# Dependencies
#
################################################################################

# Source
tpccdbg$(OBJEXT):   tpccdbg.c
tpccctx$(OBJEXT):    tpccctx.c
tpccmisc$(OBJEXT): tpccmisc.c

# Headers
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h

UTIL_OBJ_DB2 = tpccctx$(OBJEXT)

#
################################################################################
# User Targets
```

Src.Common/tpccmisc.c

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 *
 * tpccmisc.c - Miscellaneous routines
 *
 */

#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>

double current_time_ms(void);
double current_time(void);


/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
  /* use time() to get seconds */
  return(time(NULL));
}
```

```
/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
   /* gettimeofday() returns seconds and microseconds */
   /* convert to fractional seconds */
  struct timeval t;
  gettimeofday(&t,NULL);
  return (t.tv_sec + (double)t.tv_usec/(1000*1000));
}
```

dbgen/Makefile

```
################################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################
######

# Makefile - Build gendata tool
#

include $(TPCC_ROOT)/Makefile.config

#
################################################################################
# Preprocessor, Compiler and Linker Flags
#
################################################################################

INCLUDE =       -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =    $(INCLUDE) $(CFLAGS_OS) -DLINT_ARGS -DSQLA_NOLINES \
        -D$(DB2EDITION) -D$(DB2VERSION) $(CFLAGS_DEBUG)

LDFLAGS =  $(LDFLAGS_EXEC) $(LDFLAGS_LIB)

#
################################################################################
# File Collections
#
################################################################################

OBJS =     tpccrnd$(OBJEXT) \
        $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT)
OBJ_EEE =  $(TPCC_ROOT)/Src.Common/tpcclwh$(OBJEXT)

EXEC =     gendata$(BINEXT)

#
################################################################################
# End-User Targets
#
################################################################################

all:    $(EXEC)

clean:
    - $(ERASE) *$(OBJEXT) $(EXEC)
```

```
#
################################################################################
# Build Rules
#
################################################################################

.SUFFIXES:
.SUFFIXES:      $(OBJEXT) .c

# We use $@$(OBJEXT) here so that the UNIX makefiles work with both
# 'traditional' make and GNU make
$(EXEC):
    $(LD_EXEC) $(LDFLAGS) $(OBJS) $@$(OBJEXT) $(LDFLAGS_OUT)$@

#
################################################################################
# Dependencies
#
################################################################################

# Link Dependencies
gendata$(BINEXT):   $(OBJS) gendata$(OBJEXT)

# Build Dependencies
# Source
gendata$(OBJEXT):  gendata.c

# Headers
gendata.c: $(TPCC_ROOT)/include/tpccrnd.h $(TPCC_ROOT)/include/lval.h
```

dbgen/gendata.c

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * gendata.c - Generate data for TPC-C database
 *
 */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sqlutil.h>
/* UNIX named pipe support */
#include <sys/stat.h>
#include <errno.h>
#include <fcntl.h>
#include <time.h>

#include "platform.h"
#include "db2tpcc.h"
#include "tpccrnd.h"
#include "tpccmisc.h"
#include "lval.h"


/*  PROTOTYPES.  */
void gen_dist_tbl( void );
void gen_cust_tbl( void );
void gen_hist_tbl( void );
void gen_nu_ord_tbl( void );
void gen_ordr_tbl( void );
void gen_item_tbl( void );
void gen_stock_tbl( void );
void gen_ware_tbl( void );

int i, j;
double timestamp1, timestamp2, elapse;
int rc, rc1, rc2;

int using_range = 0;
int using_npipe = 0;
int using_rctload = 0;
int quiet_mode = 0;
sqlint32 ware_start=-1, ware_end=-1;

char fmtWare[]    = "%s|%s|%s|%s|%s|%s|%04.4f|%.2f|%d\n";
char fmtDist[]    = "%d|%04.4f|%.2f|%s|%s|%s|%s|%s|%d|%d\n";
char fmtItem[]    = "%s|%.2f|%s|%d|%d\n";
char fmtStock[]   = "%d|%d|%d|%d|%d|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%d|%d\n";
char fmtCust[]    =
"%d|%s|%s|%s|%s|%.2f|%s|%s|%04.4f|%s|%s|%s|%s|%s|%d|%d|%d|%.2f|%.2f|%d\n";
char fmtHist[]    = "%d|%d|%d|%d|%d|%d|%s|%.2f|%s\n";
char fmtOrdr[]    = "%d|%s|%d|%d|%d|%d|%d|%d\n";
char fmtOLine[]   = "%s|%.2f|%d|%d|%d|%s|%d|%d|%d|%d\n";
char fmtNewOrd[]  = "%d|%d|%d\n";
void InitFormatStrings(char delim);
void ScalingReport(void);

int outtype1 = 0;
int outtype2 = 0;
char *outname1 = NULL;
char *outname2 = NULL;

/*-------------------------------------------------------------------------*/
/*      main                                                               */
/*-------------------------------------------------------------------------*/
int main (int argc, char *argv[])
{
  int option = -1;
  char *delim = NULL;

  /* ********************************************************************* */
  /* Compute Warehouse Ranges                                              */
  /* ********************************************************************* */
  ware_start = 1;
  ware_end = WAREHOUSES;

  /* ********************************************************************* */
  /* Process Command Line Arguments                                        */
  /* ********************************************************************* */

  /* Valid Command Line Options
   * -------------------------
   * Table Option:             -t <table>         (-t 3 for warehouse)
   * Output Column Delimiter:  -d <char>          (-d ' ', -d '|', etc)
   * Output to File:           -f[n] <file>       (-f customer.dat)
   * Output to Pipe:           -p[n] <pipename>   (-p tpccpipe.000)
   * Warehouse Range:          -r <start> <end>   (-r 1 100)
   * Scaling Report:           -s
   * Quiet Mode:               -q
   *
   * The -f[n] and/or -p[n] options are required.
   * The -t, -d, -r, -s and -q options are optional.
   *
   * If -d is omitted, the vertical bar (pipe) symbol ('|') will be used.
   * If -r is omitted, the range [1..WAREHOUSES] will be used.
   *
   * Due to the TPC-C spec requiring that orders and orderline be
   * generated at the same time, there is an extension to the -f and -p
   * options to specify one of the two output streams for each argument.
   *
   * -f1 orders.dat -f2 orderline.dat will output to two files
   * -f1 orders.dat -p2 tpccpipe.000 will output to a file and a pipe
   *
   * -f1/-p1 specifies the destination for the orders table
   * -f2/-p2 specifies the destination for the orderline table
   *
   */

  /* Read Arguments */
  for (i=1; i<argc; i++)
  {
    if (strcmp(argv[i], "-t") == 0) {
      option = atoi(argv[i+1]);
      i++;
    } else if (strcmp(argv[i], "-r") == 0) {
      ware_start = atoi(argv[i+1]);
      ware_end = atoi(argv[i+2]);
      i += 2;
    } else if (strcmp(argv[i], "-d") == 0) {
      delim = argv[i+1];
      i++;
    } else if ((strcmp(argv[i], "-f")  == 0) ||
               (strcmp(argv[i], "-f1") == 0)) {
      outtype1 = IOH_FILE;
      outname1 = argv[i+1];
      i++;
    } else if (strcmp(argv[i], "-f2") == 0) {
      outtype2 = IOH_FILE;
      outname2 = argv[i+1];
      i++;
    } else if ((strcmp(argv[i], "-p")  == 0) ||
               (strcmp(argv[i], "-p1") == 0)) {
      outtype1 = IOH_PIPE;
      outname1 = argv[i+1];
      i++;
    } else if (strcmp(argv[i], "-p2") == 0) {
      outtype2 = IOH_PIPE;
      outname2 = argv[i+1];
      i++;
    } else if (strcmp(argv[i], "-s") == 0) {
      ScalingReport();
      exit(0);
    } else if (strcmp(argv[i], "-q") == 0) {
      quiet_mode = 1;
    } else {
      fprintf(stderr, "gendata: Don't understand argument: %s\n",argv[i]);
      exit(-1);
    }
  }

  /* ********************************************************************* */
  /* Validate Command Line Arguments                                       */
  /* ********************************************************************* */

  /* Validate Table Argument */
  if (option < 3 || option > 11 || option == 10)
  {
    fprintf(stderr,"gendata: Invalid table selected: %d\n",option);
    exit(-1);
  }

  /* Validate Delimiter Argument */
  if (delim == NULL) {
    // default delimiter is used for IMPORT & LOAD, no changes neccessary
    using_rctload = 0;
  } else if (strlen(delim) == 1 && !isalnum(delim[0]) &&
             delim[0] != '.' && delim[0] != '%')
```

```
  {
    // user-supplied delimiter used for rctload
    InitFormatStrings(delim[0]);
    using_rctload = 1;
  } else {
    fprintf(stderr,"gendata: Invalid delimiter specified: %s\n",delim);
    exit(-1);
  }

  /* Validate File/Pipe Arguments */
  if (option != 9 && outtype1 > 0 && outtype2 > 0)
  {
    fprintf(stderr,"gendata: Specifying two output file/pipes allowed only when
generating\norders/orderline.\n");
    exit(-1);
  }
  if (option == 9 && ((outtype1 == 0) || (outtype2 == 0)))
  {
    fprintf(stderr,"gendata: Must specify two output file/pipes when generating
orders/orderline.\n");
    exit(-1);
  }
  if (outtype1 == 0 || outname1 == NULL || strcmp(outname1,"") == 0)
  {
    fprintf(stderr,"gendata: Invalid 1st output file/pipe specified.\n");
    exit(-1);
  }
  if (option == 9 && (outtype2 == 0 || outname2 == NULL || strcmp(outname2,"") == 0))
  {
    fprintf(stderr,"gendata: Invalid 2nd output file/pipe specified.\n");
    exit(-1);
  }
  /* Ensure O/OL flat files are opened in append mode.  This is required */
  /* because we generate O/OL concurrently.  See comments in genload.pl  */
  /* for further details on why this is neccessary.                      */
  if (option == 9)
  {
    if (outtype1 == IOH_FILE) outtype1 = IOH_FILE_APPEND;
    if (outtype2 == IOH_FILE) outtype2 = IOH_FILE_APPEND;
  }

  /* Validate Range Arguments */
  if (ware_start <= 0 || ware_start > WAREHOUSES) {
    fprintf(stderr,"gendata: Invalid range starting value: %d\n",ware_start);
     exit(-1);
  }
  if (ware_end <= 0 || ware_end > WAREHOUSES || ware_end < ware_start) {
    fprintf(stderr,"gendata: Invalid range ending value: %d\n",ware_end);
    exit(-1);
  }

  initialize_random();

  /* ********************************************************************** */
  /* Generate Data                                                          */
  /* ********************************************************************** */
  switch (option) {
   case 3: /* WAREHOUSE */
     gen_ware_tbl();
     break;
   case 4: /* DISTRICT  */
     gen_dist_tbl();
     break;
   case 5: /* ITEM  */
     gen_item_tbl();
     break;
   case 6: /* STOCK     */
     gen_stock_tbl();
     break;
   case 7: /* CUSTOMER  */
     gen_cust_tbl();
     break;
   case 8: /* HISTORY   */
     gen_hist_tbl();
     break;
   case 9: /* ORDERS + ORDER_LINE */
     gen_ordr_tbl();
     break;
   case 11: /* NEW_ORDER */
     gen_nu_ord_tbl();
     break;
   case 2:
   case 10:
   default:
     fprintf(stderr, "Error: invalid option = %d \n",(option));
     break;
  }
  return 0;
}

/*-----------------------------------------------------------------------*/
/* generate item table                                                   */
/*-----------------------------------------------------------------------*/

void gen_item_tbl( void )
{
  sqlint32  item_num = 0 ;
  sqlint32  item_im_id ;
  char  item_name[25] ;
  double  item_price ;
  char  item_data[51] ;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto item_done; }

  for(item_num = 1; item_num <= ITEMS; item_num++)
  {
    /* create image id field */
    item_im_id = rand_integer( 1, 10000 ) ;
    /* create name field */
    create_random_a_string( item_name, 14, 24);
    /* create price field */
    item_price = rand_decimal( 100, 10000, 2 ) ;
    /* create ORIGINAL field */
    create_a_string_with_original( item_data, 26, 50, 10) ;

    numBytes = sprintf(Buffer, fmtItem,
                       item_name,
                       item_price,
                       item_data,
                       item_im_id,
                       item_num);

    rc = GenericWrite(&hnd, Buffer, numBytes);
    if (rc != 0) { goto item_done; }

  } /* end for... */

  rc = GenericClose(&hnd);

item_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;

  if (rc == 0) {
    if (!quiet_mode) {
      fprintf(stdout,"\nITEM table generated in %8.2f seconds.\n\n",elapse);
      fflush(stdout);
    }
  } else {
    fprintf(stderr,"\nITEM table FAILED at (I %d) after %8.2f
seconds.\n\n",item_num,elapse);
    fflush(stderr);
  }
}

/*-----------------------------------------------------------------------*/
/* generate stock table                                                  */
/*-----------------------------------------------------------------------*/
void gen_stock_tbl( void )
{
  sqlint32 ware_num = 0 ;
  sqlint32 stock_num = 0 ;
  sqlint32 stock_quant;
  sqlint32 s_ytd;
  sqlint32 s_order_cnt, s_remote_cnt;
  char  stock_dist_01[25] ;
  char  stock_dist_02[25] ;
  char  stock_dist_03[25] ;
  char  stock_dist_04[25] ;
  char  stock_dist_05[25] ;
  char  stock_dist_06[25] ;
  char  stock_dist_07[25] ;
  char  stock_dist_08[25] ;
  char  stock_dist_09[25] ;
  char  stock_dist_10[25] ;
  char  stock_data[51] ;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto stock_done; }

  for (stock_num = 1; stock_num <= STOCK_PER_WAREHOUSE; stock_num++)
  {
    if (!quiet_mode && (stock_num%500 == 0))
    {
      fprintf(stdout, "STOCK for Item #%d\n",stock_num);
      fflush(stdout);
    }
    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
      stock_quant = rand_integer( 10, 100 ) ;
      create_random_a_string( stock_dist_01, 24, 24);
      create_random_a_string( stock_dist_02, 24, 24);
      create_random_a_string( stock_dist_03, 24, 24);
      create_random_a_string( stock_dist_04, 24, 24);
      create_random_a_string( stock_dist_05, 24, 24);
      create_random_a_string( stock_dist_06, 24, 24);
      create_random_a_string( stock_dist_07, 24, 24);
      create_random_a_string( stock_dist_08, 24, 24);
      create_random_a_string( stock_dist_09, 24, 24);
      create_random_a_string( stock_dist_10, 24, 24);

      /* create ORIGINAL field */
      create_a_string_with_original( stock_data, 26, 50, 10 );
      s_ytd = s_order_cnt = s_remote_cnt = 0;

      numBytes = sprintf(Buffer, fmtStock,
                         s_remote_cnt,
```

```
                    stock_quant,
                    s_order_cnt,
                    s_ytd,
                    stock_data,
                    stock_dist_01,
                    stock_dist_02,
                    stock_dist_03,
                    stock_dist_04,
                    stock_dist_05,
                    stock_dist_06,
                    stock_dist_07,
                    stock_dist_08,
                    stock_dist_09,
                    stock_dist_10,
                    stock_num,
                    ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto stock_done; }

      } /* end for... */
    } /* end for... */

    rc = GenericClose(&hnd);

stock_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
      if (!quiet_mode) {
        fprintf(stdout,"\nSTOCK table generated in %8.2f seconds.\n\n",elapse);
        fflush(stdout);
      }
    } else {
      fprintf(stderr,"\nSTOCK table FAILED at (S %d W %d) after %8.2f
seconds.\n\n",stock_num, ware_num, elapse);
      fflush(stderr);
    }
}

/*----------------------------------------------------------------------*/
/* generate warehouse table                                             */
/*----------------------------------------------------------------------*/
void gen_ware_tbl( void )
{
    sqlint32 ware_num = 0 ;
    char ware_name[11] ;
    char ware_street_1[21] ;
    char ware_street_2[21] ;
    char ware_city[21] ;
    char ware_state[3]  ;
    char ware_zip[10]  ;
    double ware_tax ;
    double ware_YTD ;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto ware_done; }

    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
      if (!quiet_mode && ((ware_num % 500) == 0)) {
        fprintf(stdout, "Warehouse #%d\n", ware_num);
        fflush(stdout);
      }

      create_random_a_string( ware_name,      6,10) ; /* create name */
      create_random_a_string( ware_street_1, 10,20) ; /* create street 1 */
      create_random_a_string( ware_street_2, 10,20) ; /* create street 2 */
      create_random_a_string( ware_city,     10,20) ; /* create city */
      create_random_a_string( ware_state,     2,2) ;  /* create state */
      create_random_n_string( ware_zip,       4,4) ;  /* create zip */
      strcat(ware_zip, "11111");

      ware_tax = rand_decimal(0, 2000,4);
      ware_YTD = 300000.00;

      numBytes = sprintf(Buffer, fmtWare,
                    ware_name,
                    ware_street_1,
                    ware_street_2,
                    ware_city,
                    ware_state,
                    ware_zip,
              ware_tax,
              ware_YTD,
                    ware_num);

      rc = GenericWrite(&hnd, Buffer, numBytes);
      if (rc != 0) { goto ware_done; }

    } /* end for */

    rc = GenericClose(&hnd);

ware_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
      if (!quiet_mode) {
        fprintf(stdout,"\nWAREHOUSE table generated in %8.2f seconds.\n\n",elapse);
        fflush(stdout);
      }
    } else {
      fprintf(stderr,"\nWAREHOUSE table FAILED at (W %d) after %8.2f
seconds.\n\n",ware_num, elapse);
      fflush(stderr);
    }
}

/*----------------------------------------------------------------------*/
/* generate dist table                                                  */
/*----------------------------------------------------------------------*/
void gen_dist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    char dist_name[11];
    char dist_street_1[21];
    char dist_street_2[21];
    char dist_city[21];
    char dist_state[3];
    char dist_zip[10];
    double dist_tax;
    sqlint32 next_o_id;
    double dist_YTD;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    next_o_id = CUSTOMERS_PER_DISTRICT + 1;
    timestamp1 = current_time();

    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto dist_done; }

    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
      if (!quiet_mode) {
        fprintf(stdout, "DISTRICT for Warehouse #%d\n", ware_num);
        fflush(stdout);
      }
      for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
      {
        create_random_a_string( dist_name,      6,10) ; /* create name */
        create_random_a_string( dist_street_1, 10,20) ; /* create street 1 */
        create_random_a_string( dist_street_2, 10,20) ; /* create street 2 */
        create_random_a_string( dist_city,     10,20) ; /* create city */
        create_random_a_string( dist_state,     2,2) ;  /* create state */
        create_random_n_string( dist_zip,       4,4) ;  /* create zip */
        strcat(dist_zip, "11111");
        dist_tax = rand_decimal(0, 2000,4);
        dist_YTD = 30000.00;

        numBytes = sprintf(Buffer, fmtDist,
                      next_o_id,
                      dist_tax,
                      dist_YTD,
                      dist_name,
                      dist_street_1,
                      dist_street_2,
                      dist_city,
                      dist_state,
                      dist_zip,
                      dist_num,
                      ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto dist_done; }

      } /* end for... */
    } /* end for... */

    rc = GenericClose(&hnd);

dist_done:

    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
      if (!quiet_mode) {
        fprintf(stdout,"\nDISTRICT table generated in %8.2f seconds.\n\n",elapse);
        fflush(stdout);
      }
    } else {
      fprintf(stderr,"\nDISTRICT table FAILED at (W %d D %d) after %8.2f
seconds.\n\n",ware_num,dist_num,elapse);
      fflush(stderr);
    }
}

/*----------------------------------------------------------------------*/
/* generate customer table                                              */
/*----------------------------------------------------------------------*/
void gen_cust_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    char cust_last[17];
    char cust_middle[3];
    char cust_first[17];
```

```
char cust_street_1[21];
char cust_street_2[21];
char cust_city[21];
char cust_state[3];
char cust_zip[10];
char cust_phone[17];
char cust_credit[3];
char cust_data[501];
char cust_since[27];
double cust_discount;
double cust_balance;
double cust_YTD_payment;
double cust_credit_lim;

IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];
int len, pos;

timestamp1 = current_time();

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto cust_done; }

strcpy(cust_middle, "OE");

createTimestampString(cust_since);

for (cust_num = 1; cust_num <= CUSTOMERS_PER_DISTRICT; cust_num++)
{
  if (!quiet_mode) {
    fprintf(stdout, "CUSTOMER #%d:\n", cust_num);
    fflush(stdout);
  }

  for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
  {
    for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
    {
      if (cust_num <= 1000)                    /* create last name */
        create_random_last_name( cust_last, cust_num);
      else                                     /* create last name */
        create_random_last_name( cust_last, 0);
      create_random_a_string( cust_first,     8,16) ; /* create first name */
      create_random_a_string( cust_street_1, 10,20) ; /* create street 1 */
      create_random_a_string( cust_street_2, 10,20) ; /* create street 2 */
      create_random_a_string( cust_city,     10,20) ; /* create city */
      create_random_a_string( cust_state,     2,2) ;  /* create state */
      create_random_n_string( cust_zip,       4,4) ;  /* create zip */
      strcat(cust_zip, "11111");

      /* create phone number */
      create_random_n_string( cust_phone, 16,16) ;
      if ( rand_integer( 1, 100 ) <= 10 )
        strcpy( cust_credit, "BC" ) ;
      else
        strcpy( cust_credit, "GC" ) ;

      /* create discount rate */
      cust_discount = rand_decimal(0,5000,4);

      /* create customer data */
      create_random_a_string(cust_data, 300, 500);

      /* pad customer data (only for non-rctload) */
      if (using_rctload == 0) {
        for (pos=strlen(cust_data); pos<500; pos++)
          cust_data[pos] = ' ';
        cust_data[500] = '\0';
      }

      cust_credit_lim = 50000.00;
      cust_balance = -10.00;
      cust_YTD_payment = 10.00;

      if (cust_num == 1 && dist_num == 1 && ware_num == 1)
      {
        sprintf(cust_first,"C_LAST_LOAD=%d",C_C_LAST_LOAD);
      }

      numBytes = sprintf(Buffer, fmtCust,
                         cust_num,
                         cust_state,
                         cust_zip,
                         cust_phone,
                         cust_since,
                         cust_credit_lim,
                         cust_middle,
                         cust_credit,
                         cust_discount,
                         cust_data,
                         cust_last,
                         cust_first,
                         cust_street_1,
                         cust_street_2,
                         cust_city,
                         dist_num,
                         ware_num,
                         0,
                         cust_balance,
                         cust_YTD_payment,
                         1);

      rc = GenericWrite(&hnd, Buffer, numBytes);
      if (rc != 0) { goto cust_done; }

    } /* end for district... */
  } /* end for warehouse... */
} /* end for customer... */

rc = GenericClose(&hnd);

cust_done:

timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
  if (!quiet_mode) {
    fprintf(stdout,"\nCUSTOMER table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  }
} else {
  fprintf(stderr,"\nCUSTOMER table FAILED at (W %d D %d C %d) after %8.2f
seconds.\n\n",ware_num, dist_num, cust_num, elapse);
  fflush(stderr);
}
}

/*----------------------------------------------------------------------*/
/* generate hist table                                                  */
/*----------------------------------------------------------------------*/
void gen_hist_tbl( void )
{
  sqlint32 ware_num = 0 ;
  sqlint32 dist_num = 0 ;
  sqlint32 cust_num = 0 ;
  char hist_data[25] ;
  char h_date[27] ;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto hist_done; }

  createTimestampString(h_date);

  for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
  {
    if (!quiet_mode) {
      fprintf(stdout, "HISTORY for Warehouse #%d:\n", ware_num);
      fflush(stdout);
    }
    for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
    {
      for (cust_num = 1; cust_num <= CUSTOMERS_PER_DISTRICT; cust_num++)
      {
        /* create history data */
        create_random_a_string( hist_data, 12,24) ;

        numBytes = sprintf(Buffer, fmtHist,
                           cust_num,
                           dist_num,
                           ware_num,
                           dist_num,
                           ware_num,
                           h_date,
                    10.00,
                           hist_data);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto hist_done; }

      } /* end for customer... */
    } /* end for district... */
  } /* end for warehouse... */

  rc = GenericClose(&hnd);

hist_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    if (!quiet_mode) {
      fprintf(stdout,"\nHISTORY table generated in %8.2f seconds.\n\n",elapse);
      fflush(stdout);
    }
  } else {
    fprintf(stderr,"\nHISTORY table FAILED at (W %d D %d C %d) after %8.2f
seconds.\n\n",ware_num, dist_num, cust_num, elapse);
    fflush(stderr);
  }
}

/*----------------------------------------------------------------------*/
/* generate nu_ord table                                                */
/*----------------------------------------------------------------------*/
void gen_nu_ord_tbl( void )
{
  sqlint32 ware_num = 0 ;
  sqlint32 dist_num = 0 ;
  sqlint32 nu_ord_id = 0 ;
  int nu_ord_hi ;

  IOH_NUM numBytes;
  ioHandle hnd;
```

```
   char Buffer[1024];

   /* compute maximum and minimum
      order numbers for this
      district */
   nu_ord_hi = CUSTOMERS_PER_DISTRICT - NU_ORDERS_PER_DISTRICT + 1;
   if (nu_ord_hi < 0) {
      nu_ord_hi = CUSTOMERS_PER_DISTRICT - (CUSTOMERS_PER_DISTRICT / 3) + 1;
      fprintf(stderr,"\n**** WARNING **** NU_ORDERS_PER_DISTRICT is >
CUSTOMERS_PER_DISTRICT\n");
      fprintf(stderr,"                Check the values in file lval.h\n");
      fprintf(stderr,"                Loading New-Order with 1/3 of
CUSTOMERS_PER_DISTRICT\n");
   }

   timestamp1 = current_time();

   rc = GenericOpen(&hnd, outtype1, outname1);
   if (rc != 0) { goto neword_done; }

   /* We generate in O/W/D order for non-RCT tables.  With the
    * data clustered on O_ID, this gives us good bufferpool
    * characteristics.  We also create a btree index in W/D/O
    * order, to satisfy MIN(O_ID) queries.
    *
    * For RCT tables *with* RCT Jump Cache, we *should* generate
    * the data in W/D/O order (to match the table definition.)
    * We don't since it would push schema decisions into flat file
    * generation (and I don't want to do that.)  It's just as easy
    * to sort the flat files afterwards.
    */

   for (nu_ord_id = nu_ord_hi;
        nu_ord_id <= CUSTOMERS_PER_DISTRICT;
        nu_ord_id++)
   {
      if (!quiet_mode) {
         fprintf(stdout, "NEW_ORDER for Customer #%d:\n", nu_ord_id);
         fflush(stdout);
      }
      for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
      {
         for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
         {
            numBytes = sprintf(Buffer, fmtNewOrd,
                               nu_ord_id,
                               dist_num,
                               ware_num);

            rc = GenericWrite(&hnd, Buffer, numBytes);
            if (rc != 0) { goto neword_done; }

         } /* end for... */
      } /* end for... */
   } /* end for... */

   rc = GenericClose(&hnd);

neword_done:

   timestamp2 = current_time();
   elapse = timestamp2 - timestamp1;
   if (rc == 0) {
      if (!quiet_mode) {
         fprintf(stdout,"\nNEW_ORDER table generated in %8.2f seconds.\n\n",elapse);
         fflush(stdout);
      }
   } else {
      fprintf(stderr,"\nNEW_ORDER table FAILED at (W %d D %d O %d) after %8.2f
seconds.\n\n",ware_num, dist_num, nu_ord_id, elapse);
      fflush(stderr);
   }
}

/*----------------------------------------------------------------------*/
/* generate order and order_line tables                                 */
/*----------------------------------------------------------------------*/
void gen_ordr_tbl( void )
{
   sqlint32 ware_num = 0 ;
   sqlint32 dist_num = 0 ;
   sqlint32 cust_num = 0 ;
   sqlint32 ord_num = 0 ;
   sqlint32 ordr_carrier_id;
   sqlint32 ordr_ol_cnt;
   sqlint32 oline_ol_num;
   sqlint32 oline_item_num;

   double oline_amount;
   char oline_dist_info[25];

   IOH_NUM numBytes;
   ioHandle hnd1, hnd2;
   char Buffer[1024];

   char currtmstmp[27];
   char nulltmstmp[27] = "0001-01-01 00:00:00";

   oline_dist_info[24] = '\0';

   timestamp1 = current_time();

   rc1 = GenericOpen(&hnd1, outtype1, outname1);
   if (rc1 != 0) { goto ool_done; }
   rc2 = GenericOpen(&hnd2, outtype2, outname2);
   if (rc2 != 0) { goto ool_done; }

   createTimestampString(currtmstmp);

   for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
   {
      if (!quiet_mode) {
         fprintf(stdout, "ORDERS & ORDER_LINE for Warehouse #%d\n", ware_num);
         fflush(stdout);
      }
      for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
      {
         if (!quiet_mode) {
            fprintf(stdout, "District #%d\t", dist_num);
            fflush(stdout);
         }

         seed_1_3000();

         for (ord_num = 1; ord_num <= CUSTOMERS_PER_DISTRICT; ord_num++)
         {
           if (ord_num < 2101)
              ordr_carrier_id = rand_integer( 1, 10 ) ;
           else
              ordr_carrier_id = 0;

           cust_num = random_1_3000();
           ordr_ol_cnt = rand_integer(MIN_OL_PER_ORDER,MAX_OL_PER_ORDER);

           numBytes = sprintf(Buffer, fmtOrdr,
                              cust_num,
                              currtmstmp,
                              ordr_carrier_id,
                              ordr_ol_cnt,
                              1,
                              ord_num,
                              ware_num,
                              dist_num);

           rc1 = GenericWrite(&hnd1, Buffer, numBytes);
           if (rc1 != 0) { goto ool_done; }

           for ( oline_ol_num = 1; oline_ol_num <= ordr_ol_cnt; oline_ol_num++ )
           {
              oline_item_num = rand_integer(1, ITEMS) ;
              create_random_a_string( oline_dist_info, 24, 24) ;

              numBytes = sprintf(Buffer, fmtOLine,
                                 ((ord_num < 2101) ? currtmstmp : nulltmstmp),
                                 ((ord_num < 2101) ? 0.00 : rand_decimal(1,999999,2)),
                                 oline_item_num,
                                 ware_num,
                                 5,
                                 oline_dist_info,
                                 ord_num,
                                 dist_num,
                                 ware_num,
                                 oline_ol_num);

              rc2 = GenericWrite(&hnd2, Buffer, numBytes);
              if (rc2 != 0) { goto ool_done; }

           } /* for order_line */
         } /* for order */
      } /* for dist */
   } /* for ware */

   rc1 = GenericClose(&hnd2);
   rc2 = GenericClose(&hnd1);

ool_done:

   timestamp2 = current_time();
   elapse = timestamp2 - timestamp1;
   if (rc1 == 0 && rc2 == 0) {
      if (!quiet_mode) {
         fprintf(stdout,"\nORDERS & ORDER_LINE tables generated in %8.2f
seconds.\n\n",elapse);
         fflush(stdout);
      }
   } else {
      fprintf(stderr,"\nORDERS & ORDER_LINE tables FAILED at (W %d D %d O %d OL %d)
after %8.2f seconds.\n\n",ware_num, dist_num, ord_num, oline_ol_num, elapse);
      fflush(stderr);
   }
}

// This routine will initalize the printf format strings and replace the
// delimiter with the one provided.  The pipe symbol is the default.
void InitFormatStrings(char delim)
{
  char *p;

  // Check if Using Default Delimiter
  if (delim == '|') return;

  // Replace Delimiters
  while (p = strchr(fmtWare,'|'))   { *p = delim; }
  while (p = strchr(fmtDist,'|'))   { *p = delim; }
  while (p = strchr(fmtItem,'|'))   { *p = delim; }
  while (p = strchr(fmtStock,'|'))  { *p = delim; }
  while (p = strchr(fmtCust,'|'))   { *p = delim; }
  while (p = strchr(fmtHist,'|'))   { *p = delim; }
  while (p = strchr(fmtOrdr,'|'))   { *p = delim; }
  while (p = strchr(fmtOLine,'|'))  { *p = delim; }
```

```
  while (p = strchr(fmtNewOrd,'|')) { *p = delim; }
}

void ScalingReport(void)
{
   /* Print Scaling Values */
   fprintf(stdout,"Scaling Values in Use\n");
   fprintf(stdout,"-------------------------------\n");
   fprintf(stdout,"Warehouses:            %d\n", WAREHOUSES);
   fprintf(stdout,"Districts/Warehouse:   %d\n", DISTRICTS_PER_WAREHOUSE);
   fprintf(stdout,"Customers/District:    %d\n", CUSTOMERS_PER_DISTRICT);
   fprintf(stdout,"Items:                 %d\n", ITEMS);
   fprintf(stdout,"Stock/Warehouse:       %d\n", STOCK_PER_WAREHOUSE);
   fprintf(stdout,"Min Order Lines/Order: %d\n", MIN_OL_PER_ORDER);
   fprintf(stdout,"Max Order Lines/Order: %d\n", MAX_OL_PER_ORDER);
   fprintf(stdout,"New Orders/District:   %d\n", NU_ORDERS_PER_DISTRICT);
   fprintf(stdout,"-------------------------------\n");
}
```

dbgen/tpccrnd.c

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpccrnd.c - Random generation functions for TPC-C
 *
 */

#include <stdio.h>
#include <string.h>
#include <math.h>
#include "db2tpcc.h"
#include "tpccmisc.h"
#include "lval.h"

static char tbl_cust[CUSTOMERS_PER_DISTRICT];

static char alnum[] =
   "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";

static char *last_name_parts[] =
{
   "BAR",
   "OUGHT",
   "ABLE",
   "PRI",
   "PRES",
   "ESE",
   "ANTI",
   "CALLY",
   "ATION",
   "EING"
};

/*
*****************************************************************************
*   rand_integer
*
*   create a uniform random numeric value of type integer, of random
*   value between lo and hi. Number is NOT placed in BUFFER, and IS
*   simply RETURNED.
*
*   Routine RETURNS the VALUE.
*
*   parameters
*   ----------
*   lo end of acceptable value range
*   hi end of acceptable value range
*
*   output
*   ------
*   random integer value  RETURNED
*
*****************************************************************************
*/

int rand_integer ( int val_lo, int val_hi )
{
  return((random()%(val_hi-val_lo+1))+val_lo);
}

/*
*****************************************************************************
*   rand_decimal
*
*   create a uniform random numeric value of type double, of random
*   value between lo and hi with val_dec fractional digits.
*   Number is NOT placed in BUFFER, and IS simply RETURNED.
*
*   Routine RETURNS the VALUE.
*
*   parameters
*   ----------
*   lo end of acceptable value range
*   hi end of acceptable value range
*   number of fractional digits
*
*   output
*   ------
*   random double value  RETURNED
*
*****************************************************************************
*/

double rand_decimal ( int val_lo, int val_hi, int val_dec )
{
  return(rand_integer(val_lo,val_hi)/pow(10.0,(double)val_dec));
}

/*
*****************************************************************************
*   seed_1_3000
*
*
*****************************************************************************
*/

void seed_1_3000( void )
{
  int i;

  for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++) {
    tbl_cust[i] = 0;
  }
}

/*
*****************************************************************************
*   random_1_3000
*
*
*****************************************************************************
*/

int random_1_3000( void )
{
  static int i;
  static int x;

  x = rand_integer(0, CUSTOMERS_PER_DISTRICT - 1);

  for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++)
  {
    if (tbl_cust[x] == 0)
    {
      tbl_cust[x] = 1;
      return(x+1);
    } else {
      x++;
    }
    if (x == CUSTOMERS_PER_DISTRICT)
      x=0;
  }

  printf("\nfatal error in random_1_3000 \n");
  abort();
}


/*
*****************************************************************************
*   initialize_random
*****************************************************************************
*/

void initialize_random(void)
{
  int t = current_time();

  srand(t);
  srandom(t);
}


/*
*****************************************************************************
*   create_random_a_string
*
*   create a random alphanumeric string, of random length between lo and
*   hi and place them in designated buffer. Routine returns the actual
*   length.
*
*   parameters
*   ----------
*   lo end of acceptable length range
*   hi end of acceptable length range
*
*   output
*   ------
*   actual length
*   random alphanumeric string
*
*****************************************************************************
*/

int create_random_a_string( char *out_buffer, int length_lo, int length_hi )
{
  int i, actual_length ;
```

```
  actual_length = rand_integer( length_lo, length_hi ) ;

  for (i = 0; i < actual_length; i++ )
  {
    out_buffer[i] = alnum[rand_integer( 0, 61 )] ;
  }
  out_buffer[actual_length] = '\0' ;

  return (actual_length);
}


/*
*******************************************************************************
*   create_random_n_string
*
*   create a random numeric string, of random length between lo and
*   hi and place them in designated buffer. Routine returns the actual
*   length.
*
*   parameters
*   ----------
*   lo end of acceptable length range
*   hi end of acceptable length range
*
*   output
*   ------
*   actual length
*   random numeric string
*
*******************************************************************************
*/

int create_random_n_string( char *out_buffer, int length_lo, int length_hi )
{
  int i, actual_length ;

  actual_length = rand_integer( length_lo, length_hi ) ;

  for (i = 0; i < actual_length; i++ )
  {
    out_buffer[i] = (char)rand_integer( 48,57 ) ;
  }
  out_buffer[actual_length] = '\0' ;

  return (actual_length);
}

/*
*******************************************************************************
*   NUrand_val
*
*   create a non-uniform random numeric value of type integer, of random
*   value between lo and hi. Number is NOT placed in BUFFER, and IS
*   simply RETURNED.
*
*   Routine RETURNS the VALUE.
*
*   parameters
*   ----------
*   lo end of acceptable value range
*   hi end of acceptable value range
*
*   output
*   ------
*   random integer value  RETURNED
*
*******************************************************************************
*/

int NUrand_val ( int A, int x, int y, int C )
{
  return((((rand_integer(0,A)|rand_integer(x,y))+C)%(y-x+1))+x);
}

/*
*******************************************************************************
*   create_a_string_with_original
*
*   create a random alphanumeric string, of random length between lo and
*   hi and place them in designated buffer. Routine returns the actual
*   length.
*
*   the word "ORIGINAL" is placed at a random location in the buffer at
*   random, for a given percent of the records.
*
*   percent_to_set must be an integer value from 0 to 100.
*   if 0, no records will be set. If 100, all records will be set.
*
*   CANNOT USE ON STRINGS OF LENGTH LESS THAN 8 ! LOWER LIMIT MUST BE >
8 !
*
*   parameters
*   ----------
*   lo end of acceptable length range
*   hi end of acceptable length range
*   percentage of records to set to ORIGINAL
*
*   output
*   ------
*   actual length
*   random alphanumeric string with the word "ORIGINAL" is placed at a
*       random location
*
*******************************************************************************
*/

int create_a_string_with_original( char *out_buffer, int length_lo,
                                   int length_hi, int percent_to_set )
{
  int actual_length, start_pos ;

  actual_length = create_random_a_string( out_buffer, length_lo, length_hi ) ;

  if ( rand_integer( 1, 100 ) <= percent_to_set )
  {
    start_pos = rand_integer( 0, actual_length-8 ) ;
    strncpy(out_buffer+start_pos,"ORIGINAL",8) ;
  }

  return (actual_length);
}


/*******************************************************************************
 *
 * create_random_last_name
 *
 * parameters:
 *    out_buffer - target buffer for the generated last name
 *
 * description:
 *    create_random_last_name generates a random number from 0 to 999
 *    inclusive. a random name is generated by associating a random string
 *    with each digit of the generated number. the three strings are
 *    concatenated to generate the name
 *
 *******************************************************************************/

int create_random_last_name(char *out_buffer, int cust_num)
{
 int random_num;

 if (cust_num == 0)
   random_num = NUrand_val( A_C_LAST, 0, 999, C_C_LAST_LOAD );
 else
   random_num = cust_num - 1;

 strcpy(out_buffer, last_name_parts[random_num / 100]);
 random_num %= 100;
 strcat(out_buffer, last_name_parts[random_num / 10]);
 random_num %= 10;
 strcat(out_buffer, last_name_parts[random_num]);

 return(strlen(out_buffer));
}
```

include/db2tpcc.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * db2tpcc.h - Macros and Miscellany
 *
 */

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

/* ************************************************************************** */
/* Transaction Return Codes (s_transtatus)                                    */
/* ************************************************************************** */

#define INVALID_ITEM            100
#define TRAN_OK                 0
#define FATAL_SQLERROR          -1

/* ************************************************************************** */
/* Definition of Unused and Bad Items                                         */
/* ************************************************************************** */
/* Define unused item ID to be 0. This allows the SUT to determine the     */
/* number of items in the order as required by 2.4.1.3 and 2.4.2.2 since   */
/* the assumption that any item with OL_I_ID = 0 is unused will be true.   */
/* This in turn requires that the value used for an invalid item is        */
/* equal to ITEMS + 1.                                                     */
/* ************************************************************************** */

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****************************************************************************/
/* NURand Constants                                                          */
```

```
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.          */
/* Analysis indicates that a C_LAST delta of 85 is optimal.             */
/*****************************************************************************/
#define C_C_LAST_RUN  88
#define C_C_LAST_LOAD 173
#define C_C_ID    319
#define C_OL_I_ID   3849
#define A_C_LAST   255
#define A_C_ID    1023
#define A_OL_I_ID   8191

/*****************************************************************************/
/* Transaction Type Identifiers                                              */
/*****************************************************************************/

#define CLIENT_SQL   0
#define NEWORD_SQL   1
#define PAYMENT_SQL  2
#define ORDSTAT_SQL  3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   in_items_struct {
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad1[3];
  } in_item[15];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  s_O_OL_CNT;    /* init by SUT */
  int16_t  s_all_local;
  int16_t  duplicate_items;
};

struct out_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   items_struct {
    float  s_I_PRICE;
    float  s_OL_AMOUNT;
    int16_t  s_S_QUANTITY;
    int16_t  pad2;
    char     s_I_NAME[25];
    char     s_brand_generic;
  } item[15];
  float  s_W_TAX;
  float  s_D_TAX;
  float  s_C_DISCOUNT;
  float  s_total_amount;
  int32_t  s_O_ID;
  int16_t  s_O_OL_CNT;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_LAST[17];
  char     s_C_CREDIT[3];
  char     s_O_ENTRY_D_time[27];

};

struct in_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  float  s_H_AMOUNT;
  int32_t  s_W_ID;
  int32_t  s_C_W_ID;
  int32_t  s_C_ID;
  int16_t  s_C_D_ID;
  int16_t  s_D_ID;
  char     s_C_LAST[17];
};

struct out_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  double  s_C_CREDIT_LIM;
  double  s_C_BALANCE;
  float  s_C_DISCOUNT;
  int32_t  s_C_ID;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_W_STREET_1[21];
  char     s_W_STREET_2[21];
  char     s_W_CITY[21];
  char     s_W_STATE[3];
  char     s_W_ZIP[10];
  char     s_D_STREET_1[21];
  char     s_D_STREET_2[21];
  char     s_D_CITY[21];
  char     s_D_STATE[3];
  char     s_D_ZIP[10];
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_C_STREET_1[21];
  char     s_C_STREET_2[21];
  char     s_C_CITY[21];
  char     s_C_STATE[3];
  char     s_C_ZIP[10];
  char     s_C_PHONE[17];
  char     s_C_CREDIT[3];
  char     s_C_DATA[201];
  char     s_H_DATE_time[27];
  char     s_C_SINCE_time[27];
};

struct in_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  pad1[3];
  char     s_C_LAST[17];
};

struct out_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  double  s_C_BALANCE;
  int32_t  s_C_ID;
  int32_t  s_O_ID;
  int16_t  s_O_CARRIER_ID;
  int16_t  s_ol_cnt;
  int16_t  pad1[2];
  struct   oitems_struct {
    double  s_OL_AMOUNT;
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad2;
    char     s_OL_DELIVERY_D_time[27];
  } item[15];
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_O_ENTRY_D_time[27];
  int16_t  pad3[2];
};

struct in_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_W_ID;
  int16_t  s_O_CARRIER_ID;
};

struct out_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_O_ID[10];
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

struct in_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_threshold;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
};

struct out_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_low_stock;
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

/* ************************************************************************ */
/* Transaction Prototypes                                                   */
/* ************************************************************************ */

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/* ************************************************************************ */
/* DB2 Connect/Disconnect & Thread Context Wrappers                         */
/* ************************************************************************ */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);
```

```
#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H
```

include/lval.h

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 41600
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H
```

include/platform.h

```
/*********************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*********************************************************************/

/*
 * platform.h - Platform Isolation Layer
 *
 */

#ifndef __PLATFORM_H
#define __PLATFORM_H

/* ********************************************************************/
/* Generic Macros                                                     */
/* ********************************************************************/
#define GEN_ERRCODE  errno

/* ********************************************************************/
/* Windows I/O Macros                                                 */
/* ********************************************************************/

/* ********************************************************************/
/* UNIX I/O Macros                                                    */
/* ********************************************************************/
#include <fcntl.h>

#define IOH_INIT(hnd, type, name)                                  \
  hnd->fd = -1;                                                \
  hnd->type = type;                                            \
  hnd->name = name;

#define IOH_CREATE(hnd)                                            \
  if (hnd->type == IOH_PIPE) {                                 \
    rc = mkfifo(hnd->name, 0666);                              \
  } else {                                                     \
    rc = 0;                                                    \
  }

#define IOH_OPEN(hnd)                                              \
  if (hnd->type == IOH_FILE_APPEND) {                          \
    hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_APPEND, 0666);      \
  } else {                                                     \
    hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_TRUNC, 0666);       \
  }                                                            \
  if (hnd->fd == -1) {                                         \
    rc = -1;                                                   \
  } else {                                                     \
    rc = 0;                                                    \
  }

#define IOH_WRITE(hnd, buff, num, num2)                            \
  rc = write(hnd->fd, buff, num);                              \
  if (rc >= 0) {                                               \
    num2 = rc;                                                 \
    rc = 0;                                                    \
  }

#define IOH_FLUSH(hnd)  rc = 0;
#define IOH_CLOSE(hnd) rc = close(hnd->fd);
#define IOH_DELETE(hnd)    if (hnd->type == IOH_PIPE) { rc = unlink(hnd->name); }

typedef unsigned int IOH_NUM;
typedef int IOH_HND;

/* ********************************************************************/
/* UNIX Semaphore Macros                                              */
/* ********************************************************************/
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

union semun {
  int val;
  struct semid_ds *buf;
  unsigned short int *array;
} semUnion;

struct sembuf semBuf;

#define SEM_HANDLE int

#define SEM_INIT(hnd, x, name)                                     \
  if ( (hnd = semget(IPC_PRIVATE, 1 , IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR |
S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH)) == -1)                    \
    API_ERROR(__LINE__, "semget", (rc=GEN_ERRCODE));               \
  semUnion.val = x;                                            \
  if ( semctl(hnd, 0, SETVAL, semUnion) < 0 )                  \
    API_ERROR(__LINE__, "semctl SETVAL", (rc=GEN_ERRCODE));

#define SEM_WAIT(hnd)                                              \
  semBuf.sem_num = 0;                                          \
  semBuf.sem_op = -1;                                          \
  semBuf.sem_flg = SEM_UNDO;                                   \
  if ( semop(hnd, &semBuf, 1) < 0 )                            \
    API_ERROR(__LINE__, "semop wait", (rc=GEN_ERRCODE));

#define SEM_FREE(hnd)                                              \
  semBuf.sem_num = 0;                                          \
  semBuf.sem_op = 1;                                           \
  semBuf.sem_flg = SEM_UNDO;                                   \
  if ( semop(hnd, &semBuf, 1) < 0 )                            \
    API_ERROR(__LINE__, "semop free", (rc=GEN_ERRCODE));

#define SEM_DESTROY(hnd)                                           \
  if ( semctl(hnd, 0, IPC_RMID, 0) )                           \
    API_ERROR(__LINE__, "semctl IPC_RMID", (rc=GEN_ERRCODE));
```

```
/* ********************************************************************/
/* Common I/O Macros and Definitions                                  */
/* ********************************************************************/
#define IOH_FILE 1
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3

#define IOH_ERRMSG(hnd, msg)                                       \
  if (rc != 0) {                                               \
    fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", GEN_ERRCODE, msg,      \
            hnd->fd, hnd->type, hnd->name);                    \
    return rc;                                                 \
  }

struct _ioh {
  IOH_HND fd;
  int type;
  char *name;
};

typedef struct _ioh ioHandle;

/* ********************************************************************/
/* Generic I/O Routine Prototypes                                     */
/* ********************************************************************/
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes);
int GenericClose(ioHandle *hnd);

/* ********************************************************************/
/* Generic I/O Routines                                               */
/* ********************************************************************/
int GenericOpen(ioHandle *hnd, int type, char *name)
{
  int rc = 0;

  IOH_INIT(hnd, type, name)

  IOH_CREATE(hnd)
  IOH_ERRMSG(hnd, "creating")

  IOH_OPEN(hnd)
  IOH_ERRMSG(hnd, "opening")

  return rc;
}

int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes)
{
  int rc = 0;
  int numBytesWritten = -1;

  IOH_WRITE(hnd, Buffer, numBytes, numBytesWritten)
  IOH_ERRMSG(hnd, "writing")
  if (numBytes != numBytesWritten) {
    fprintf(stderr, "Truncated data writing to fd %d (%d, %s)\n", hnd->fd, hnd->type, hnd-
>name);
    rc = -1;
  }
  return rc;
}

int GenericClose(ioHandle *hnd)
{
  int rc = 0;

  IOH_FLUSH(hnd)
  IOH_ERRMSG(hnd, "flushing")
```

```
   IOH_CLOSE(hnd)
   IOH_ERRMSG(hnd, "closing")

   IOH_DELETE(hnd)
   IOH_ERRMSG(hnd, "deleting")

   return rc;
}

#endif // __PLATFORM_H
```

include/tpccmisc.h

```
/*************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*************************************************************************/

/*
 * tpccmisc.h - Miscellanouse Routines
 *
 */

#ifndef __TPCCMISC_H
#define __TPCCMISC_H


extern double current_time_ms(void);
extern double current_time(void);

#include <time.h>
#define createTimestampString(buf)                    \
{                                 \
  time_t now;                                   \
  struct tm *tm;                                \
  time(&now);                                    \
  tm = localtime(&now);                              \
  sprintf(buf,                                 \
        "%4.4d-%2.2d-%2.2d %2.2d:%2.2d:%2.2d",       \
        tm->tm_year + 1900, tm->tm_mon + 1, tm->tm_mday, \
        tm->tm_hour, tm->tm_min, tm->tm_sec);        \
}

#endif // __TPCCMISC_H
```

include/tpccrnd.h

```
/*************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*************************************************************************/

/*
 * tpccrnd.h - Random generation functions for TPC-C
 *
 */

#ifndef __TPCCRND_H
#define __TPCCRND_H

void initialize_random(void);
int rand_integer( int val_lo, int val_hi ) ;
double rand_decimal( int val_lo, int val_hi, int val_dec ) ;
int NUrand_val( int A, int val_lo, int val_hi, int C ) ;

void seed_1_3000( void );
int random_1_3000( void );

int create_random_a_string( char *out_buffer,
                        int length_lo,
                        int length_hi );
int create_random_n_string( char *out_buffer,
                        int length_lo,
                        int length_hi );
int create_a_string_with_original( char *out_buffer,
                            int length_lo,
                            int length_hi,
                            int percent_to_set ) ;
int create_random_last_name(char *out_buffer, int cust_num);

#endif // __TPCCRND_H
```

tpccenv.sh

```
#########################################################################
#####
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2006
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#########################################################################
######

#
# tpccenv.sh - UNIX Environment Setup
#

# The Kit Version
export TPCC_VERSION=CK060815

# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}

# The OS being used (i.e. "UNIX", "LINUX", "WINDOWS")
export PLATFORM=LINUX

# The type of make command and slash used by the OS.
# (i.e. UNIX - "/", WINDOWS - "\").
# These are referenced all over the kit.
export SLASH="/";
export MAKE=make

# Specifies whether or not to use dari stored proc's for the TPC-C driver.  Set to either
DARIVERSION or NONDARI;
#export TPCC_SPTYPE=NOSP
#export TPCC_SPTYPE=SPGENERAL2
export TPCC_SPTYPE=SPGENERAL
#export TPCC_SPTYPE=DARI2SQLDA

export DB2VERSION=v8

# The schema name is typically the SQL autorization ID (or username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}

# DB2 EE/EEE Configuration
export DB2EDITION=EE
#export DB2EDITION=EEE
export DB2NODE=0
export DB2NODES=1;       # set to the number of nodes you have.  Set to 1 for EE.

# TPCC General Configuration
export TPCC_DBNAME=TPCC
export TPCC_ROOT=${HOME}/tpc-c.ibm
export TPCC_SQLLIB=${HOME}/sqllib
export TPCC_RUNDATA=${HOME}/tpccdata

# TPCC Debug Configuration
# This is the path where all error and debug logs are placed.
# To get debugging from within the stored procedures, you must
# set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
export TPCC_DEBUGDIR=/tmp

# Specifies where stored procedures should be placed and if they should
# be fenced.
export TPCC_SPDIR=${TPCC_SQLLIB}/function
export TPCC_FENCED=NO
```

12 Appendix D: Pricing

March 13, 2007

IBM Corporation
Ms. Celia Schreiber
xSeries Performance

Dear Celia:

The table shown below lists the U.S. pricing for DB2 9 Data server product that has been used in the TPC-C Benchmark.

All prices shown are in U.S. Dollars.

DB2 Enterprise Server Edition (ESE)	VUs	Reference Price per value unit	Total Reference price
DB2 Enterprise Proc 9 Lic/1 year Maintenance	800	270.59	216,472
SW Maintenance Renewal - 1 year	1600	12.89	20,624
		Sub-total reference price for DB2 ESE:	**230,136**
		TOTAL REFERENCE PRICE:	**237,096**

Any and all prices herein are suggested prices only and are subject to change at IBM's sole discretion. Products listed herein are subject to withdrawal or modification by IBM at any time at IBM's sole discretion.

Sincerely,

Richard Hughes
IBM Sales & Distribution, Software Sales
Americas Sales Executive DB2 and Informix
212-493-2065
rhughes@us.ibm.com

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

February 13, 2007

IBM Corporation
Chris King
3079 Cornwallis Road
Durham, NC 27709

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars ($).

Part Number	Description	Unit Price	Quantity	Price
254-00170	Microsoft Visual C++ .Net Standard 2003 *Full License* *No Discounts Applied*	$109	1	$109
N/A	Microsoft Problem Resolution Services *Professional Support* *(1 Incident)*	$245	1	$245

All products are currently orderable through Microsoft's normal distribution channels. A list of Microsoft's resellers can be found at http://www.microsoft.com/products/info/render.aspx?view=22&type=mnp&content=22/licensing>/b>

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at $245 per call.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCchki0721300006118.
Please include this Reference ID in any correspondence regarding this price quote.

| Home | About Us | Customer Support | View Cart | Log On

CDW | Mac Warehouse

Search | Advanced Search

Mac PC All

Brands Hardware Software Networking Accessories Services Product Finders 800.750.4239

Filter YOUR Search Results

Keyword: 430446 ☒

Show ready to ship products only Hide Product images

Sort By: **BEST MATCH** | GROUP | BRAND | LOWEST PRICE | HIGHEST PRICE

Showing 1 - 1 of **1** Products

Your selected filters have been applied to the results displayed on this page.

To modify your results, click the appropriate criteria line item or click ☒ to remove the entire line.

Compare

	Product Details	CDW#	Availability	Advertised Price
	**Linksys ProConnect KVM switch - 2 ports** KVM switch - 2 ports - 1 local user **MFG#: KVM2KIT**	430446	In Stock	$39.99

Compare

Showing 1 - 1 of **1** Products

Take Our Survey
Was our search engine helpful?

Home > My Shopping Cart

MY SHOPPING CART

My Wish Lists | Print Cart | Email Cart

Update Qtys | Remove Selected | Move Selected To...

	Qty.	Product Description	Savings	Total Price
	1	AMC CC5E-B14B 14 ft. Cat 5E Blue Cat 5E Blue Cable - OEM Item #: N82E16812105305 Return Policy: Standard Return Policy		$2.99

Subtotal: $2.99

Calculate Shipping

Zip Code: UPS Guaranteed 3 Day Service GO

Shipping: $0.00

Redeem Gift Certificates

Claim Code:

Security Code: Redeem

Gift Certificates: $0.00

Apply Promo Code

Promo Code: Apply

Promo Code: $0.00

Grand Total:* $2.99

* Above total does not include shipping or taxes. Please input zip code to calculate your grand total.

Having problems with your cart? Click here to empty your cart and start over.

Click here to view important shipping information.

Continue Shopping | Checkout »

ORDER TRACKING | CLEARANCE | SHOPPING CART

SEARCH: Go PHONE ORDERS: 800.287.2323

HOME | ABOUT US | PRODUCT RETURN | CUSTOMER SERVICE | CONTACT US | PHONE ORDERS | REBATES

Home> Networking > Switches

Select a Brand GO

- AUDIO
- AUTO ELECTRONICS
- CAMCORDERS
- CAMERAS
- CASES
- CELLULAR
- COMPUTER SYSTEM
- CONTROLLERS
- DVD & CD
- FAX MACHINES
- GAMING
- GPS NAVIGATION
- HEADSETS
- HEALTH AND BEAUTY
- HOME AND GARDEN
- INPUT DEVICES
- MEMORY
- MODEMS
- MONITORS
- MOTHERBOARDS
- MP3
- NETWORKING
- PDA HANDHELDS
- POWER
- PRINTERS
- PROCESSORS
- PROJECTORS
- RADIOS
- ROUTERS
- SCANNERS
- SHAVERS

D-LINK DGS-1024D 24-PORT 10/100/1000 UNMANAGED RACKMOUNTABLE DESKTOP SWITCH

See Full Description

Item is brand new and in Stock

Usually ships in 1-2 Business Days

Buy with Confidence

- 2 million satisfied customers since 1993
- 13 years of internet sales
- 100% customer satisfaction guarantee
- All items factory fresh with a full USA warranty
- 30 day money back guarantee

Selling Elsewhere for : ~~$284.99~~

Our Price Only: $229.99

Buy Now

Tell a friend

Save $50 at CompUPlus.com

FEATURED ACCESSORIES

Customers who purchased this also bought:

Cables

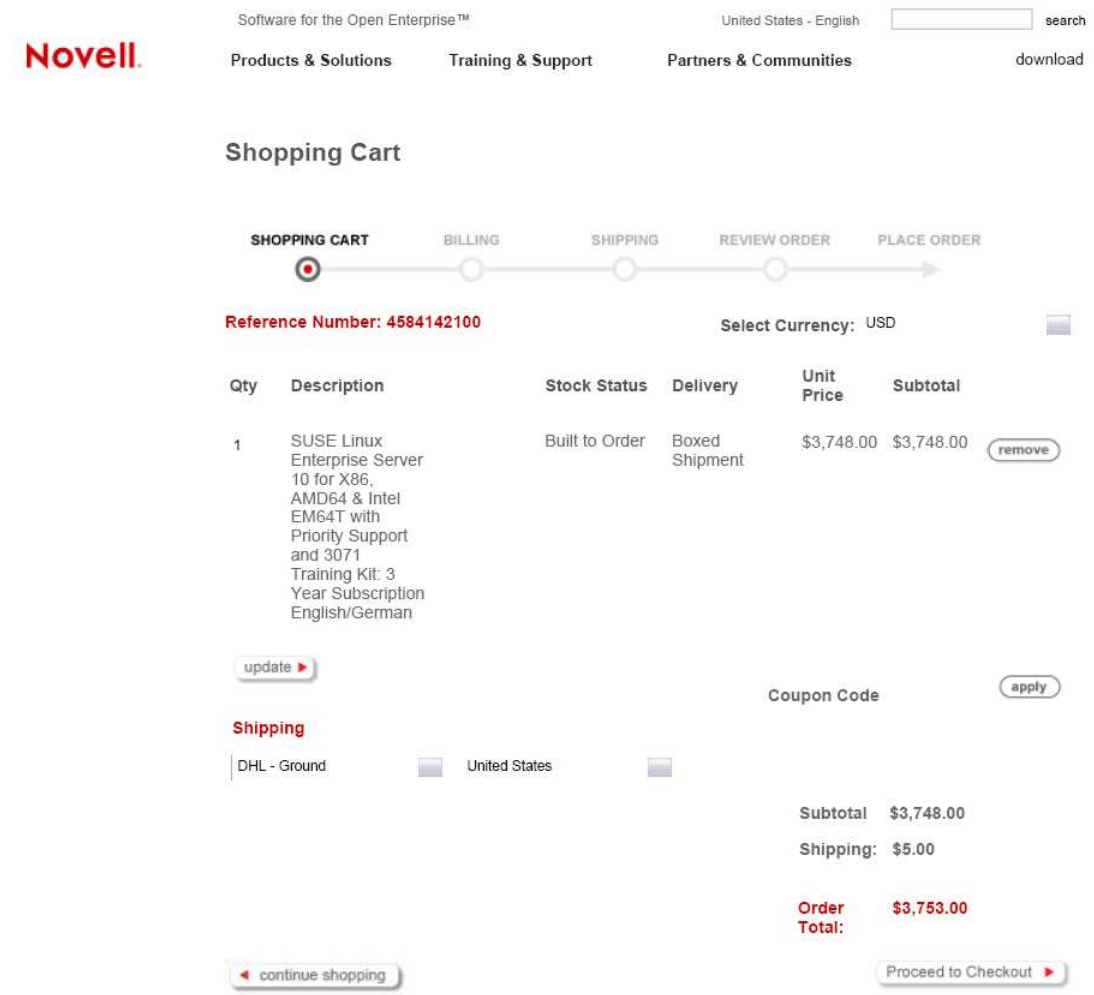
Category 5 E Gray Networking Patch Cable - 3 Foot
$3.25 More Info

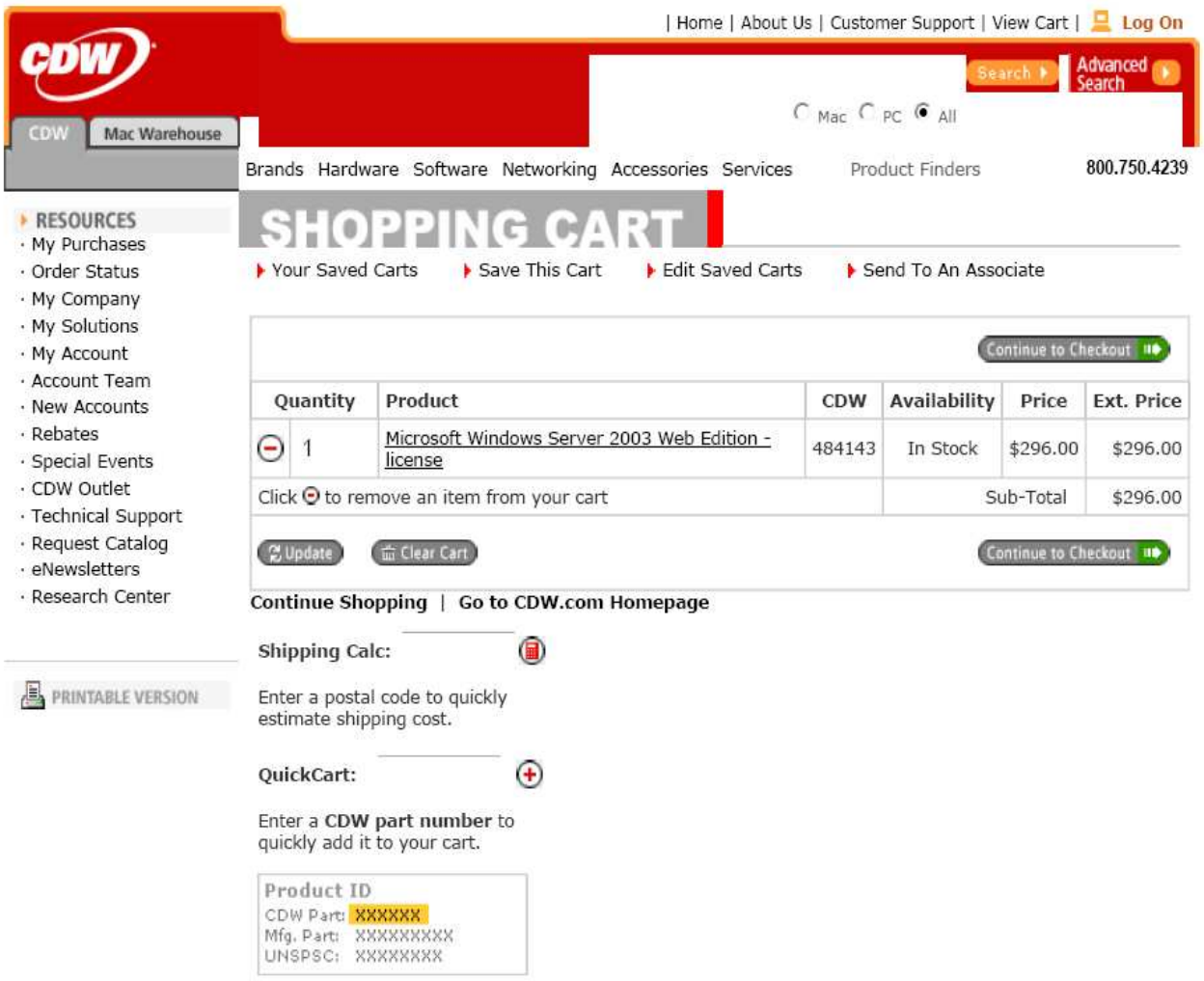
Category 5 E Gray Networking Patch Cable - 14 Foot
$4.99 More Info

Category 5 E Gray Networking Patch Cable - 7 Foot
$4.99 More Info

MORE Cables

http://www.compuplus.com/i-D-Link-DGS-1024D-24-Port-101001000-Unmanaged-Rackmountable-Des... 3/13/2007


Software for the Open Enterprise™
United States - English
search
Novell.
Products & Solutions
Training & Support
Partners & Communities
download
Shopping Cart
SHOPPING CART
BILLING
SHIPPING
REVIEW ORDER
PLACE ORDER
Reference Number: 4584142100
Select Currency: USD
Qty
Description
Stock Status
Delivery
Unit Price
Subtotal
1
SUSE Linux Enterprise Server 10 for X86, AMD64 & Intel EM64T with Priority Support and 3071 Training Kit: 3 Year Subscription English/German
Built to Order
Boxed Shipment
$3,748.00
$3,748.00
remove
update
Coupon Code
apply
Shipping
DHL - Ground
United States
Subtotal $3,748.00
Shipping: $5.00
Order Total: $3,753.00
continue shopping
Proceed to Checkout
PLEASE NOTE: If the product you have ordered is In Stock, your order will ship within 2 business days.
DHL - Ground is only available for US lower 48 States
- For UPS Ground shipments, please allow 5 - 7 business days for delivery.
DHL - Air - Next Day Air is only available for US lower 48 States
DHL - Air - Worldwide Express is available for all international shipments
- For DHL - Air - Worldwide Express, please allow 2 - 4 business days for delivery.
International Shipments: Shipping charges do not include any Brokerage Fees, Customs Fees or Taxes that you may be charged.


| Home | About Us | Customer Support | View Cart | Log On
CDW
Mac Warehouse
Search
Advanced Search
Mac PC All
Brands Hardware Software Networking Accessories Services
Product Finders
800.750.4239
RESOURCES
· My Purchases
· Order Status
· My Company
· My Solutions
· My Account
· Account Team
· New Accounts
· Rebates
· Special Events
· CDW Outlet
· Technical Support
· Request Catalog
· eNewsletters
· Research Center
PRINTABLE VERSION
SHOPPING CART
Your Saved Carts
Save This Cart
Edit Saved Carts
Send To An Associate
Continue to Checkout
Quantity
Product
CDW
Availability
Price
Ext. Price
1
Microsoft Windows Server 2003 Web Edition - license
484143
In Stock
$296.00
$296.00
Click to remove an item from your cart
Sub-Total
$296.00
Update
Clear Cart
Continue to Checkout
Continue Shopping | Go to CDW.com Homepage
Shipping Calc:
Enter a postal code to quickly estimate shipping cost.
QuickCart:
Enter a CDW part number to quickly add it to your cart.
Product ID
CDW Part: XXXXXX
Mfg. Part: XXXXXXXXX
UNSPSC: XXXXXXXX