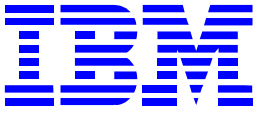
IBM System x3850 M2

Using
DB2 9.5 Enterprise Edition
and
Red Hat Enterprise Linux Advanced Platform 5

TPC Benchmark™ C
Full Disclosure Report



First Edition October 15, 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

Red Hat and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc.

First Edition: October 15, 2007

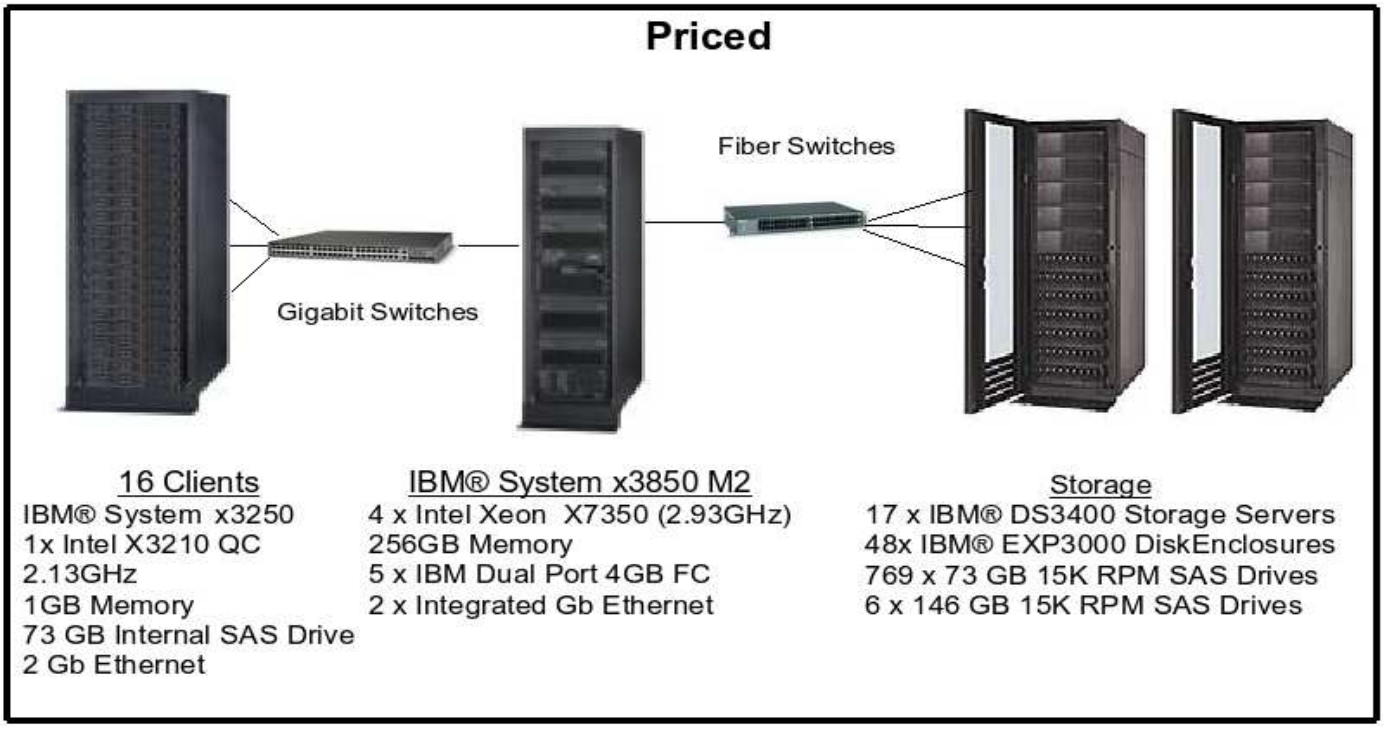
Request for additional copies of this document should be sent to the following address:

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

	IBM System x3850 M2 **DB2® 9.5**	**TPC-C Rev. 5.9** **Report Date:** **October 15, 2007**

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$1,337,458 USD	**516,752**	$2.59 USD	**March 14, 2008**

Database Processors/Cores/Threads	Database Manager	Operating System	Other Software	No. Users
4/16/16 2.93 GHz Intel Xeon X7350	DB2 9.5	RHEL 5	Microsoft Visual C++ Microsoft COM+	420,160



System Components	Each of the 16 Clients		Server	
	Quantity	Description	Quantity	Description
Processors/Cores/ Threads	1/4/4	2.13GHz 8MB L2 Xeon 3210 Quad Core	4/16/16	2.93 GHz Intel Xeon X7350
Memory	1	1 GB	32	8 GB
Disk Controllers	1	SAS	1 5 16 1	Integrated SAS 4Gb dual-port FC Adapters IBM DS3400 Controllers for Data IBM DS3400 Controller for Log
Disk Drives	1	73 GB	769 6	73 GB 15K RPM SAS 146 GB 15K RPM SAS
Total Storage		1168 GB		57,013 GB
Terminals	1	System Console	1	System Console

IBM Corporation	IBM System x3850 M2 c/s DB2 9.5				TPC-C Revision 5.9 Report Date: October 15, 2007		
Description	**Part Number**	**Third Party Brand**	**Pricing**	**Unit Price**	**Quantity**	**Extended Price**	**3-Yr. Maint. Price**
Server Hardware							
IBM System x3850 M2 (2 x Intel Xeon Processor X7350 with 2.93GHz/2x4MB L2 Cache, 4 Memory Cards, 8 x 1GB DIMM, onboard SAS enablement key, cables)	7141-4RU	IBM	1a	19,249	1	19,249	
Intel Xeon Processor X7350 (2.93GHz/1066MHz FSB/2x4MB L2)	44E4243	IBM	1a	6,269	2	12,538	
16GB (2x8GB) 667MHz PC2-5300 ECC DDR2 SDRAM DIMM	43V7356	IBM	1a*	30,000	16	480,000	
IBM T115 15-inch TFT Display	494215U	IBM	1a	209	1	209	
IBM Preferred Pro USB Keyboard	40K9584	IBM	1b	29	1	29	
IBM 3-Button Optical Mouse - Black - USB	40K9201	IBM	1b	19	1	19	
ServicePac for 3-Year 24x7x4 Support (x3850 M2)	96P2688	IBM	1b	3,390	1		3,390
ServicePac for 3-Year 24x7x4 Support (Display)	30L9183	IBM	1b	90	1		90
					Subtotal	512,044	3,480
Server Storage							
IBM 4-Gbps FC Dual-Port PCI-E HBA	39R6527	IBM	1b	1,899	5	9,495	
IBM System Storage DS3400 Express	1726-42E	IBM	1b	8,749	17	148,733	
IBM 3M SAS cable	39R6531	IBM	1b	135	96	12,960	
IBM System Storage EXP3000	1727-01X	IBM	1b	3,199	48	153,552	
IBM Hot-Swap 3.5 inch 73.4GB 15K SAS HDD	40K1043	IBM	1b	309	769	237,621	
IBM Hot-Swap 3.5 inch 146GB 15K SAS HDD	40K1044	IBM	1b	549	6	3,294	
IBM TotalStorage SAN32M-2 Express Model	202632E	IBM	1b	10,490	2	20,980	
8-Port FlexPort Expansion Kit (1 per Switch)	22R5906	IBM	1b	4,720	2	9,440	
IBM S2 42U Standard Rack	93074RX	IBM	1b	1,489	4	5,956	
DS3400 Software Feature Pack	42C2143	IBM	1b	995	16	15,920	
DS3000 EXP3000 Expansion License	39R6537	IBM	1b	1,995	16	31,920	
ServicePac for 3-Year 24x7x4 Support (DS3400)	44J8073	IBM	1b	1,300	17		22,100
ServicePac for 3-Year 24x7x4 Support (EXP3000)	41L2768	IBM	1b	760	48		36,480
ServicePac for 3-Year 24x7x4 Support (SAN32M-2)	41E9167	IBM	1b	3,300	2		6,600
ServicePac for 3-Year 24x7x4 Support (Rack)	41L2760	IBM	1b	300	4		1,200
					Subtotal	649,871	66,380
Server Software							
DB2 ESE 9.5		IBM	2	278.52	800	222,816	
Extended Systems - SW License and Maintenance 12 Months							
SW Maintenance Renewal - 1 Year		IBM	2	13.27	1600		21,232
Red Hat Enterprise Linux Advanced Platform, Premium (Unlimited Sockets)		Red Hat	3	2,499	3	7,497	
					Subtotal	230,313	21,232
Client Hardware							
Linksys ProConnect KVM Switch - 2 ports (2 spares)	430446		6	42	3	126	
IBM System x3250 with Quad-Core Intel Xeon Processor X3210 (2.13GHz) and 2x512MB Memory	43658BU	IBM	1b	2,105	16	33,680	
73.4GB 15K SAS Hot-Swap Drive	40K1043	IBM	1b	309	16	4,944	
ServicePac for 3-Year 24x7x4 Support (x3250)	40M7080	IBM	1b	219	16		3,504
					Subtotal	38,750	3,504
Client Software							
Microsoft Windows Server 2003 Web Edition with COM+	484143	Microsoft	6	296	16	4,736	
Microsoft Visual C++ Professional 6.0		Microsoft	8	200	1	200	
Microsoft Problem Resolution Services		Microsoft	4	245	1		245
					Subtotal	4,936	245
Network Components							
D-LINK DGS-1024D 24-Port 10/100/1000 Switch (2 spares)	DGS1024D		5	203	3	609	
Ethernet Cable (2 spares)	CC5E-B14B		7	3	20	60	
					Subtotal	669	
					Total	1,436,583	94,841
Large Purchase Discount (See Note 1b.)	25.46%		1			193,966	

Pricing: 1a and 1b - IBM - 1-888-SHOP-IBM, ext. 5821; 2 - IBM; 3 - Red Hat; 4 - Microsoft; 5 - compuplus.com; 6 - CDW.com; 7 - newegg.com; 8 - pricegrabber.com	**Three-Year Cost of Ownership USD:**	$1,337,458
Note 1b: Discount based on IBM Direct guidance applies to all line items where Pricing=1b. Pricing is for this system or one of similar size.	**tpmC:**	516,752
* This component is not immediately orderable. See the FDR for more information.	**$ USD/tpmC:**	$2.59
Note 2: Pricing for DB2 9.5 is based on Value Units (VUs) as shown in the price quote in Appendix D.		
Audited by Francois Raab, InfoSizing, Inc. (www.sizing.com)		

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. Thank you.

Numerical Quantities Summary for the IBM System x3850 M2

MQTH, computed Maximum Qualified Throughput: 516,752 tpmC

Response Times (in seconds)	90th %	Average	Maximum
New Order	1.80	0.59	4.60
Payment	1.80	0.57	5.63
Order-Status	1.80	0.59	4.57
Delivery (interactive)	1.02	0.34	3.16
Delivery (deferred)	0.31	0.21	2.41
Stock-Level	1.80	0.58	4.36
Menu	1.02	0.34	3.44

Response time delay added for emulated components was 0.1 seconds

Transaction Mix, in percent of total transactions	Percent
New Order	44.95%
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	18.02/120.32
Payment	3.00/0.00	3.00/12.04	3.02/120.32
Order-Status	2.00/0.00	2.00/10.04	2.02/100.32
Delivery	2.00/0.00	2.00/5.04	2.02/50.32
Stock-Level	2.00/0.00	2.00/5.04	2.02/50.32

Test Duration

Ramp-up Time	1 hour 28 minutes
Measurement interval	2 hours 00 minutes
Transactions during measurement interval (all types)	137,952,403

Checkpoints

Number of checkpoints	N/A
Checkpoint interval	N/A

Table of Contents

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.9 dated December, 2006, for measurements on the IBM System x3850 M2. The software used on the IBM System x3850 M2 includes Red Hat Enterprise Linux Advanced Platform 5 operating system and DB2 9.5 data server. Microsoft COM+ is used as the transaction manager.

IBM System x3850 M2

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM System x3850 M2	DB2 9.5	Red Hat Enterprise Linux Advanced Platform 5

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
$1,337,458 USD	516,752	$2.59 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 5.9 in September 2007.

This is the full disclosure report for benchmark testing of the IBM System x3850 M2 and DB2 9.5 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.*

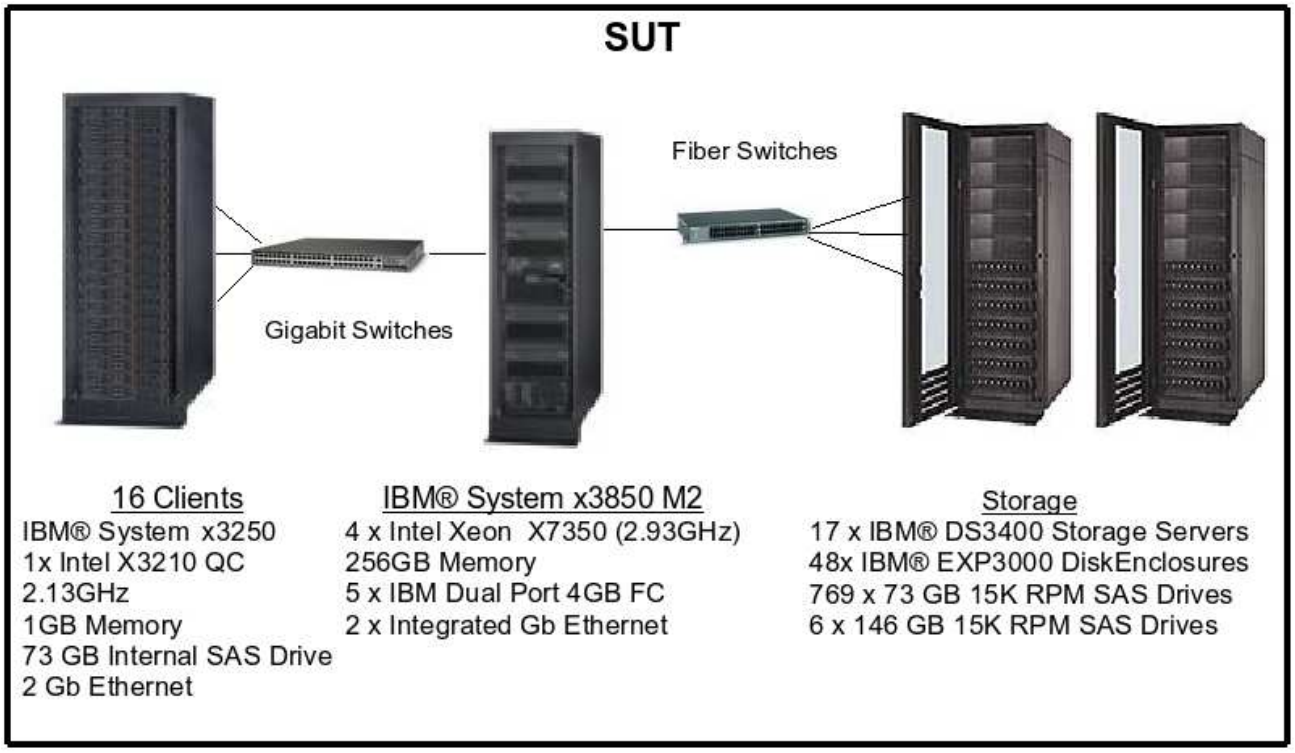
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

0.4. Configuration Diagrams

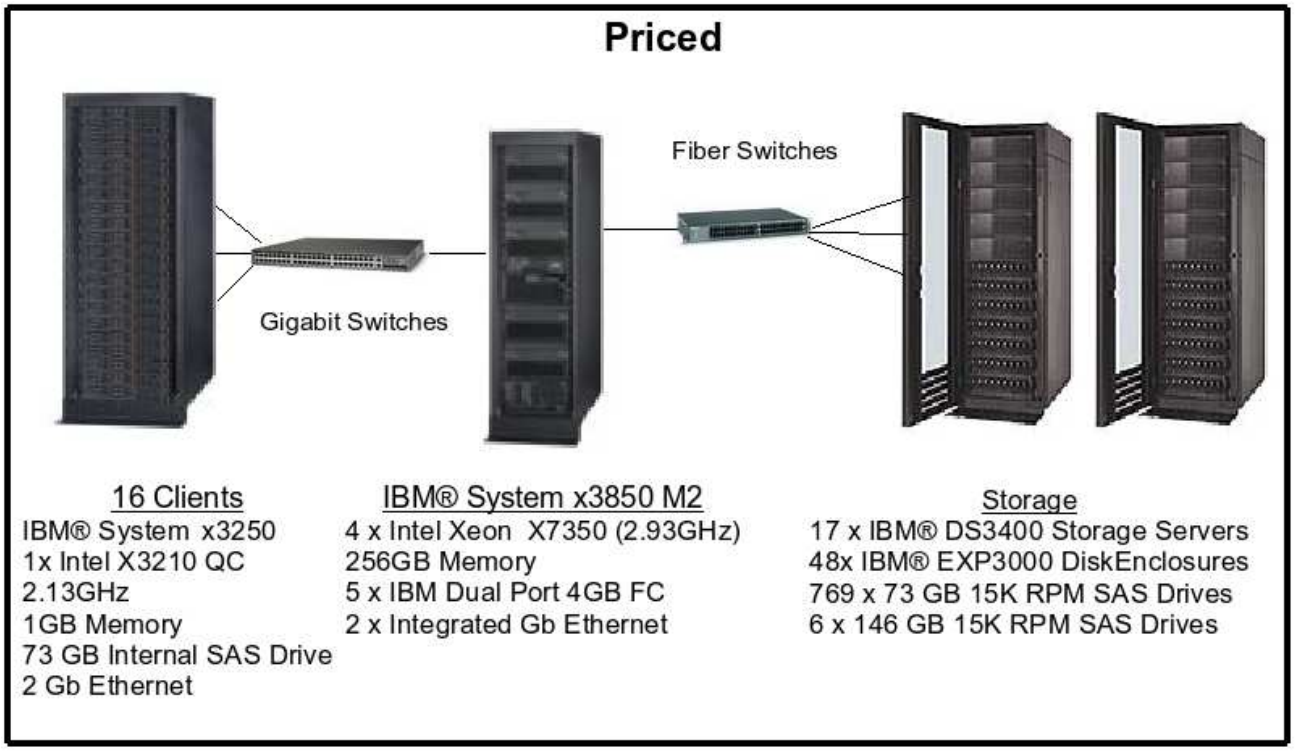
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *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 x3850 M2 Benchmark Configuration



IBM System x3850 M2 Priced Configuration



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 1313 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 System x3250 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 x3850 M2**
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	60.00%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.95%
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 and Log Fast Write Cache:

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.

Instantaneous Interruption, Memory Failure, and Loss of Power:

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 5 minutes.
3. The system was powered off, which removed power from all system components, including memory.
4. The system was powered back on and DB2 was allowed to recover.
5. 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.
6. 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	42,016
District	420,160
Customer	1,260,480,000
History	1,260,480,000
Orders	1,260,480,000
New Order	378,144,000
Order Line	12,604,768,567
Stock	4,201,600,000
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.

4.3. 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 (5+p) disk array, with 6 physical disks. Each physical disk has a capacity of 146.8 GB. The total Raid5 capacity available is 681.15 GB.

There are 32 Logical Disks (LD) for the tables.

There are 4 dual port storage adapters for the tables. Each dual port adapter is assigned 8 LDs.

Each LD is configured as a RAID0 disk array, with 24 physical disks.

There are a total of 768 data disks and each physical disk has a capacity of 73.4 GB.

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

Partition#	Blocks	Container Usage
1	7968	Warehouse
2	8001	District
3	7969	Item
4		Extended partition
5	46564308	Stock
6	33736456	Customer
7	2409694	Customer Index
8	3212966	History
9	2409660	Orders
10	2409682	Order Index
11	53014454	Order Line
12	803176	New OrdersA
13	803162	New OrdersB
14	1562505934	Not used for data tables

Tablespaces are laid out to use LDs as follows:

Warehouse	32 tablespaces each using 1 LD
District	32 tablespaces each using 1 LD
Item	One tablespace using 32 LDs
Stock	32 tablespaces each using 1 LD
Customer	32 tablespaces each using 1 LD
Customer Index	32 tablespaces each using 1 LD
History	32 tablespaces each using 1 LD
Orders	32 tablespaces each using 1 LD
Order Index	32 tablespaces each using 1 LD
Order Line	32 tablespaces each using 1 LD
New OrdersA	32 tablespaces each using 1 LD
New OrdersB	32 tablespaces each using 1 LD

4.4. 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 9.5. 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.

4.5. 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
LD1 – LD8 Partition 2	FC1	Raw41 –raw48	ts_dis_01-ts_dis_08

LD9 – LD16 Partition 2	FC2	Raw49 –raw56	ts_dis_09-ts_dis_16
LD17 – LD24 Partition 2	FC3	Raw57 –raw64	ts_dis_17-ts_dis_24
LD25 – LD32 Partition 2	FC4	Raw65 –raw72	ts_dis_25-ts_dis_32
LD1 – LD8 Partition 3	FC1	Raw81 –raw88	ts_item
LD9 – LD16 Partition 3	FC2	Raw89 –raw96	ts_item
LD17 – LD24 Partition 3	FC3	Raw97 –raw104	ts_item
LD25 – LD32 Partition 3	FC4	Raw105 –raw112	ts_item
LD1 – LD8 Partition 5	FC1	Raw121 –raw128	ts_stock_01-ts_stock_08
LD9 – LD16 Partition 5	FC2	Raw129 –raw136	ts_stock_09-ts_stock_16
LD17 – LD24 Partition 5	FC3	Raw137 –raw144	ts_stock_17-ts_stock_24
LD25 – LD32 Partition 5	FC4	Raw145 –raw152	ts_stock_25-ts_stock_32
LD1 – LD8 Partition 6	FC1	Raw161 –raw168	ts_customer_01-ts_customer_08
LD9 – LD16 Partition 6	FC2	Raw169–raw176	ts_customer_09-ts_customer_16
LD17 – LD24 Partition 6	FC3	Raw177 –raw184	ts_customer_17-ts_customer_24
LD25 – LD32 Partition 6	FC4	Raw185 –raw192	ts_customer_25-ts_customer_32
LD1 – LD8 Partition 7	FC1	Raw201 –raw208	is_customer_01-is_customer_08
LD9 – LD16 Partition 7	FC2	Raw209 –raw216	is_customer_09-is_customer_16
LD17 – LD24 Partition 7	FC3	Raw217 –raw224	is_customer_17-is_customer_24
LD25 – LD32 Partition 7	FC4	Raw225 –raw232	is_customer_25-is_customer_32
LD1 – LD8 Partition 8	FC1	Raw241 –raw248	ts_history_01-ts_history_08
LD9 – LD16 Partition 8	FC2	Raw249 –raw256	ts_history_09-ts_history_16
LD17 – LD24 Partition 8	FC3	Raw257 –raw264	ts_history_17-ts_history_24
LD25 – LD32 Partition 8	FC4	Raw265 –raw272	ts_history_25-ts_history_32
LD1 – LD8 Partition 9	FC1	Raw281 –raw288	ts_order_01-ts_order_08
LD9 – LD16 Partition 9	FC2	Raw289 –raw296	ts_order_09-ts_order_16
LD17 – LD24 Partition 9	FC3	Raw297 –raw304	ts_order_17-ts_order_24
LD25 – LD32 Partition 9	FC4	Raw305 –raw312	ts_order_25-ts_order_32
LD1 – LD8 Partition 10	FC1	Raw321 –raw328	is_order_01-is_order_08
LD9 – LD16 Partition 10	FC2	Raw329 –raw336	is_order_09-is_order_16
LD17 – LD24 Partition 10	FC3	Raw337 –raw344	is_order_17-is_order_24
LD25 – LD32 Partition 10	FC4	Raw345 –raw352	is_order_25-is_order_32
LD1 – LD8 Partition 11	FC1	Raw361 –raw368	ts_orderline_01-ts_orderline_08
LD9 – LD16 Partition 11	FC2	Raw369 –raw376	ts_orderline_09-ts_orderline_16
LD17 – LD24 Partition 11	FC3	Raw377 –raw384	ts_orderline_17-ts_orderline_24

LD25 – LD32 Partition 11	FC4	Raw385 –raw392	ts_orderline_25-ts_orderline_32
LD1 – LD8 Partition 12	FC1	Raw401 –raw408	ts_newordA_01-ts_newordA_08
LD9 – LD16 Partition 12	FC2	Raw409 –raw416	ts_newordA_09-ts_newordA_16
LD17 – LD24 Partition 12	FC3	Raw417 –raw424	ts_newordA_17-ts_newordA_24
LD25 – LD32 Partition 12	FC4	Raw425 –raw432	ts_newordA_25-ts_newordA_32
LD1 – LD8 Partition 13	FC1	Raw441 –raw448	ts_newordB_01-ts_newordB_08
LD9 – LD16 Partition 13	FC2	Raw449 –raw456	ts_newordB_09-ts_newordB_16
LD17 – LD24 Partition 13	FC3	Raw457 –raw464	ts_newordB_17-ts_newordB_24
LD25 – LD32 Partition 13	FC4	Raw465 –raw472	ts_newordB_25-ts_newordB_32

Table 4-2: IBM System x3850 M2 Data Distribution Benchmark Configuration

4.6. 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

60-Day Space Computation					
All data sizes in MB unless otherwise stated					
Warehouses	42,016				
Measured TpmC	516,752				
Table	**Rows**	**Table**	**Index**	**5% Space**	**Total Space**
Warehouse	42,016	24	0	1	25
District	420,160	64	0	3	67
Item	100,000	10	0	1	11
Stock	4,201,600,000	1,367,772	0	68,389	1,436,161
Customer	1,260,480,000	984,796	60,648	52,272	1,097,716
New-Order	378,144,000	29,040	0	1,452	30,492
Orders	1,260,480,000	48,286	35,232	0	83,518
Order-Line	12,604,800,000	844,473	0	0	844,473
History	1,260,480,000	77,616	0	0	77,616
Additional Overhead		726,658			726,658
Free Space	134,436				
Dynamic Space	970,374		30 Minute log Computations		
Static Space	3,326,361		Log Written (KB)		36,741,067
Daily Growth	190,953		New-Order Txns		15,502,560
Daily Spread	0		Log Written per New-Order (		2.37
Data Storage Requirement					
60 Days (MB)	14,783,549				
60 Days (GB)	14,437				
Log Storage Requirement					
8 Hours (GB)	560.62				
Disk Sizing					
	Formatted	**SUT**		**Priced**	
Disk Type	**Capacity (GB)**	**# of Disks**	**Capacity (GB)**	**# of Disks**	**Capacity (GB)**
DB DS3400 RAID0	73.40	768	56,371	768	56,371
LOG DS3400 RAID5	146.80	6	734	6	734
OS DS3400 RAID0	73.40	1	73	1	73
Total Capacity					57,179

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 %	1.80	1.80	1.80	1.02/0.31	1.80	1.02
Average	0.59	0.57	0.58	0.34/0.21	0.58	0.34
Maximum	4.60	5.63	4.57	3.16/2.41	4.36	3.44
Think Times						
Minimum	0	0	0	0	0	N/A
Average	12.04	12.04	10.04	5.04	5.04	N/A
Maximum	120.32	120.32	100.32	50.32	50.32	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.

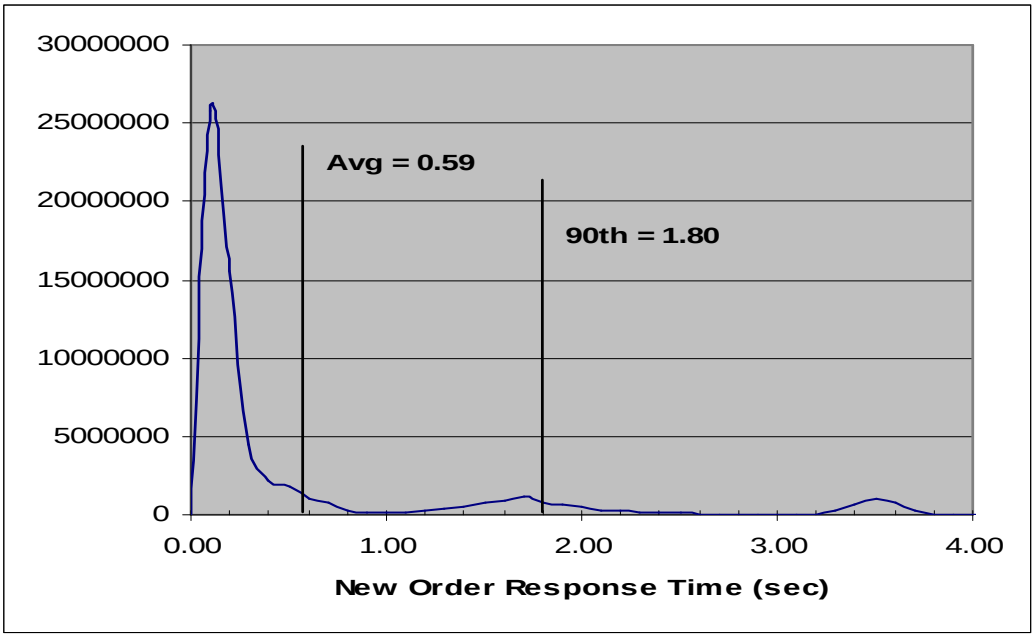


Figure 5-1: New-Order Response Time Distribution

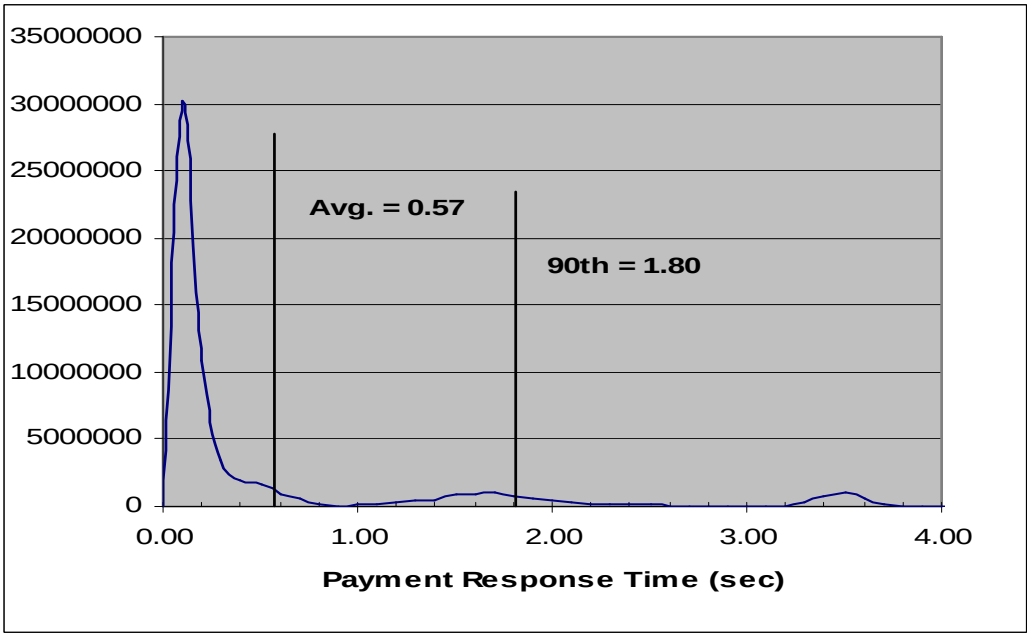


Figure 5-2: Payment Response Time Distribution

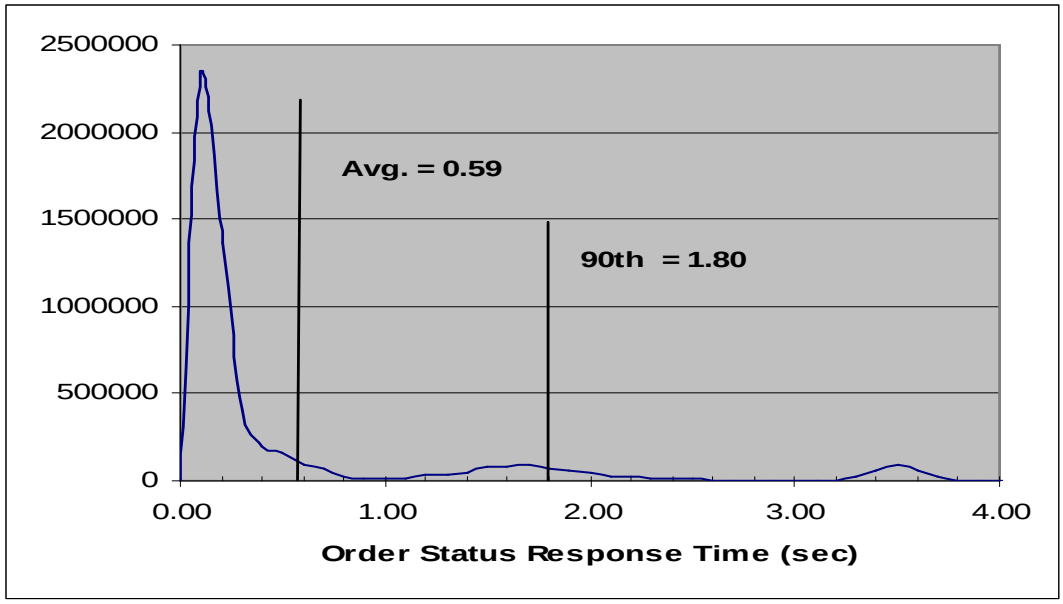


Figure 5-3: Order-Status Response Time Distribution

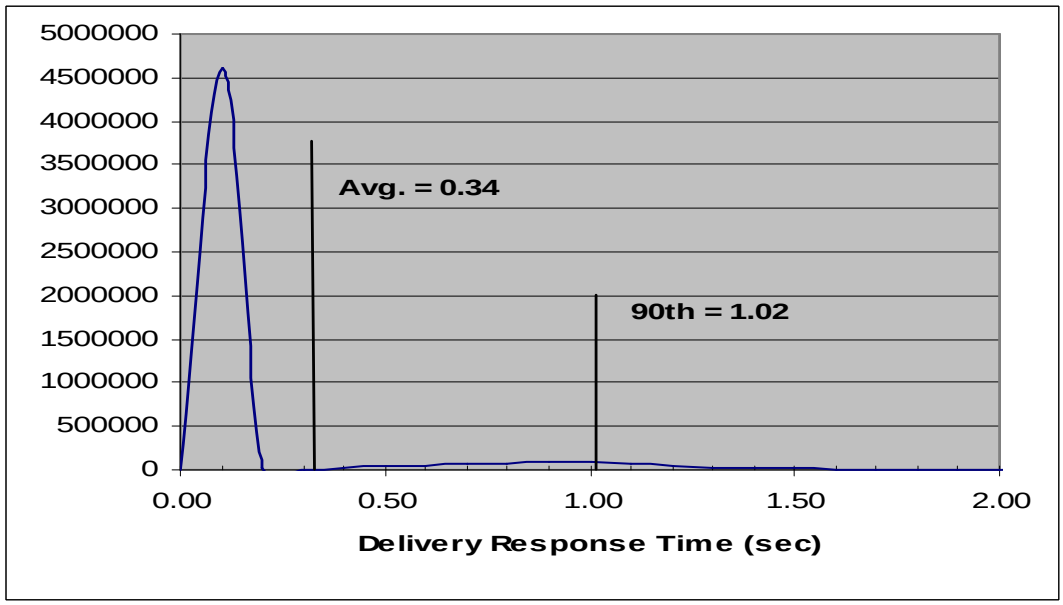


Figure 5-4: Delivery (Interactive) Response Time Distribution

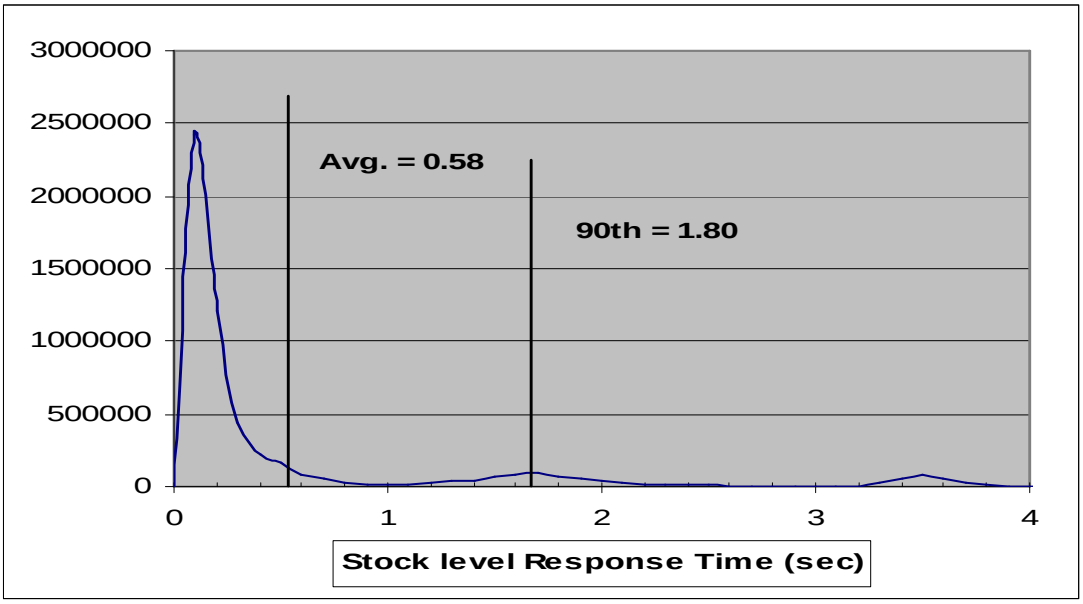


Figure 5-5: Stock Level Response Time Distribution

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.

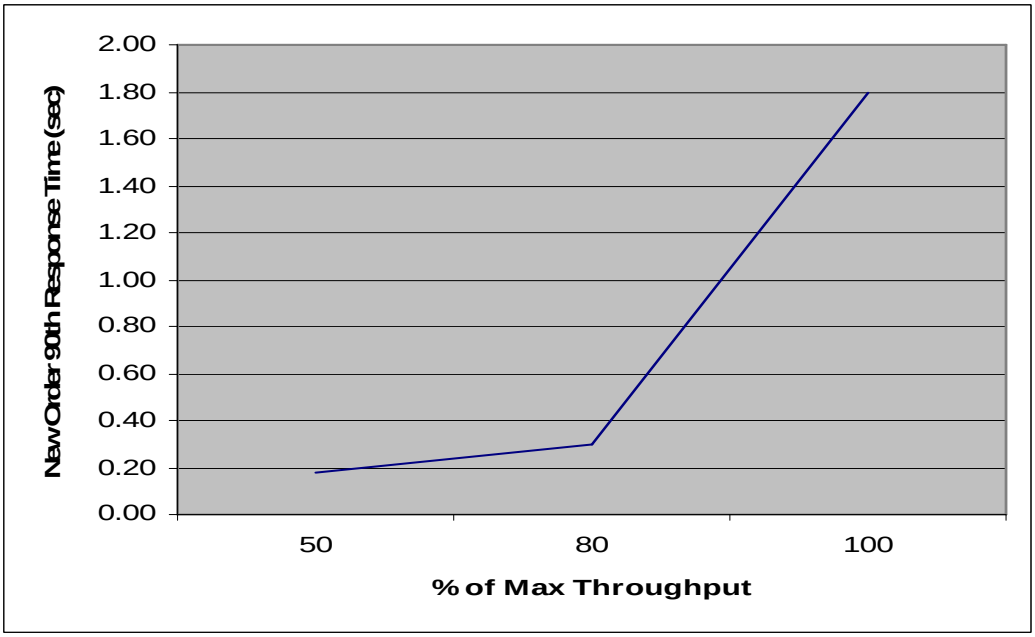


Figure 5-6: New-Order Response Time vs. Throughput

5.5. Think Time Frequency Distribution

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

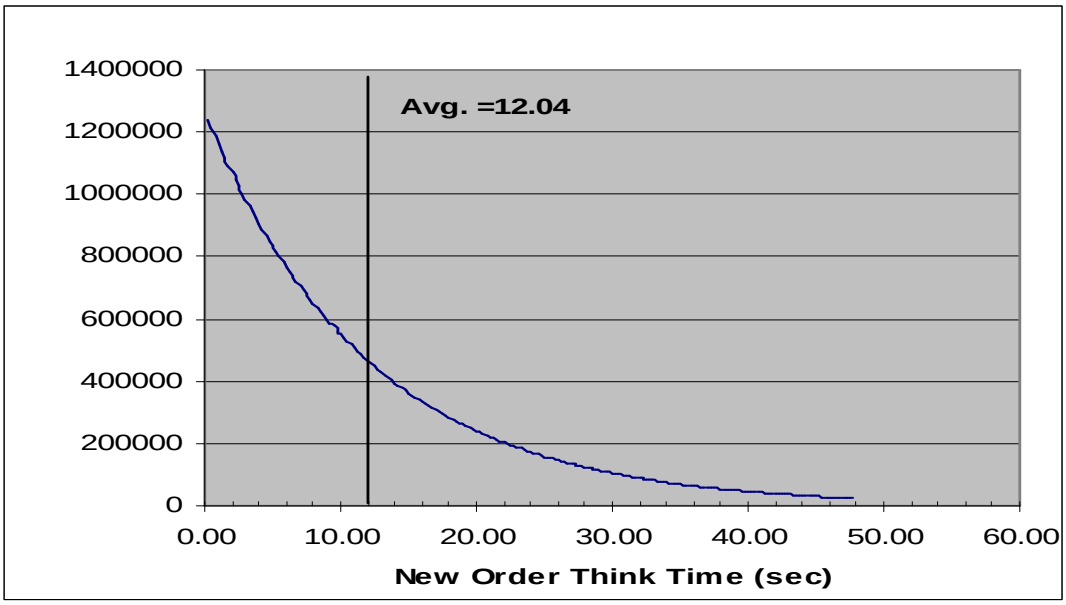


Figure 5-7: New-Order Think Time Distribution

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. 42,016 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.

Discounts are based on US list prices and for similar quantities and configurations. A discount of 25.45% 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 14, 2008.

Description	Part Number	Order Date/ Availability	Order Method	Price Verification
IBM System x3850 M2 with Intel Xeon Processor X7350	7141-4RU	10-16-07/ 12-7-07	See Note 1	See Note 1
Intel Xeon Processor X7350	44E4243	10-16-07/ 12-7-07	See Note 1	See Note 1
16GB (2x8GB) 667MHz PC2-5300 ECC DDR2 SDRAM	43V7356	1-29-08/ 3-14-08	See Note 1	See Note 2
Note 1: IBM – 1-888-746-7426, ext. 5821				
Note 2: This component is not immediately orderable. For price verification before order date, call 1-888-746-7426, ext. 5821.				

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 x3850 M2	516,752	$1,337,458 USD	$2.59 USD	March 14, 2008

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 CERTIFIED AUDITOR

Test Sponsors:

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

Berni Schiefer
IBM SB2 Performance
8200 Warden Avenue
Markham, Ontario L6G1C7

October 12, 2007

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

Platform: IBM System x3850 M2 c/s
Operating system: Red Hat Enterprise Linux Advanced Platform 5
Database Manager: DB2 9.5 Enterprise Edition
Transaction Manager: Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	New Order 90% Response Time	**tpmC**
Server: IBM System x3850 M2				
4 x Intel Xeon X7350 (2.93GHz)	256 GB (2 x 4MB L2)	769 x 73.4 GB 15K rpm SAS 6 x 146.4 GB 15K rpm SAS	1.80 Seconds	**516,752.73**
16 Client: IBM System x3250 (each with)				
1 x Intel Xeon X3210 QC (2.13 GHz)	1 GB (8 MB L2 cache)	1 x 73.4 GB SAS	n/a	n/a

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

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

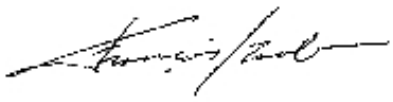
125 West Monroe Street • Colorado Springs, CO 80907 • 719 473 7363 • www.sizing.com

- 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:

None.

Respectfully Yours,



François Raab, President

125 West Monroe Street • Colorado Springs, CO 80907 • 719 473 7353 • 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.sgc

```
/*********************************************************************************
** 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 42016
#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

  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 42016
#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

```
DB2_LARGE_PAGE_MEM=DB
DB2_RESOURCE_POLICY=/home/tpcc/tpc-c.ibm/aff2_sixteen_listener_8groups.cfg
DB2_SELUDI_COMM_BUFFER=Y
DB2_USE_ALTERNATE_PAGE_CLEANING=YES
DB2_MAX_NON_TABLE_LOCKS=500
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
DB2MEMDISCLAIM=NO
DB2_COLLECT_TS_REC_INFO=false
DB2COMM=tcpip
DB2CHKPTR=OFF
```

db.cfg.out

```
     Database Configuration for Database tpcc

Database configuration release level            = 0x0c00
Database release level                          = 0x0c00

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) =
Number compatibility                            = OFF
Varchar2 compatibility                          = OFF
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

Decimal floating point rounding mode  (DECFLT_ROUNDING) = ROUND_HALF_EVEN

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) = 65336000
Database memory threshold              (DB_MEM_THRESH) = 10
Max storage for lock list (4KB)            (LOCKLIST) = 16000
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) = 65536
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
SQL statement heap (4KB)                  (STMTHEAP) = 65536
Default application heap (4KB)          (APPLHEAPSZ) = 1000
Application Memory Size (4KB)           (APPL_MEMORY) = AUTOMATIC
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                           = S0000009.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) = 1785
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
HADR peer window duration (seconds)  (HADR_PEER_WINDOW) = 0

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
Auto deletion of recovery objects    (AUTO_DEL_REC_OBJ) = OFF

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 statement statistics  (AUTO_STMT_STATS) = OFF
    Automatic statistics profiling     (AUTO_STATS_PROF) = OFF
      Automatic profile updates          (AUTO_PROF_UPD) = OFF
    Automatic reorganization             (AUTO_REORG) = OFF

Enable XML Character operations         (ENABLE_XMLCHAR) = YES
WLM Collection Interval (minutes)      (WLM_COLLECT_INT) = 0
```

dbm.cfg.out

```
        Database Manager Configuration

   Node type = Enterprise Server Edition with local and remote clients

Database manager configuration release level            = 0x0c00

CPU speed (millisec/instruction)             (CPUSPEED) = 1.810653e-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
Cluster manager                    (CLUSTER_MGR) =

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) = 2048
Audit buffer size (4KB)              (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB)  (INSTANCE_MEMORY) = 66088000
Backup buffer default size (4KB)          (BACKBUFSZ) = 1024
Restore buffer default size (4KB)         (RESTBUFSZ) = 1024

Agent stack size                  (AGENT_STACK_SZ) = 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
Agent pool size                   (NUM_POOLAGENTS) = 0
Initial number of agents in pool       (NUM_INITAGENTS) = 0
Max number of coordinating agents     (MAX_COORDAGENTS) = AUTOMATIC
Max number of client connections      (MAX_CONNECTIONS) = AUTOMATIC

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_8groups.cfg

```
<!-- This policy is valid -->
<RESOURCE_POLICY>
  <DATABASE_RESOURCE_POLICY>
    <DBNAME>tpcc</DBNAME>
    <METHOD>NODEMASK</METHOD>

    <RESOURCE_BINDING>
      <RESOURCE>0</RESOURCE>
      <DBMEM_PERCENTAGE>99.9</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>
      <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>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>5</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>13</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>21</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>29</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>37</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>45</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>53</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>6</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>14</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>22</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>30</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>38</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>46</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>54</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>7</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>15</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>23</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>31</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>39</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>47</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>55</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>8</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>16</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>24</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>32</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>40</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>48</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>56</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>9</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>17</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>25</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>33</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>41</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>49</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>57</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>10</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>18</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>26</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>34</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>42</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>50</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>58</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>11</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>19</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>27</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>35</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>43</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>51</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>59</BUFFERPOOL_ID>
      </BUFFERPOOL_BINDING>
      <BUFFERPOOL_BINDING>
        <NUM_CLEANERS>2</NUM_CLEANERS>
        <BUFFERPOOL_ID>12</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>20</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>28</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>36</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>44</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>52</BUFFERPOOL_ID>
        <BUFFERPOOL_ID>60</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
"MaxConcurrency"=dword:00000320
"MaxPoolThreads"=dword:000000c8
```

tcpip_parameters_registery.reg

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="client8"
"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"="client8"
"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
"MaxUserPort"=dword:00009c40

[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,00,44,00,36,00,33,00,44,00,39,00,41,00,\
  33,00,35,00,2d,00,42,00,33,00,33,00,41,00,2d,00,34,00,32,00,41,00,32,00,2d,\
  00,41,00,44,00,42,00,41,00,2d,00,38,00,42,00,43,00,35,00,37,00,45,00,32,00,\
  31,00,38,00,31,00,31,00,31,00,7d,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,00,45,00,33,\
  00,42,00,37,00,46,00,44,00,44,00,33,00,2d,00,31,00,44,00,45,00,32,00,2d,00,\
  34,00,45,00,43,00,45,00,2d,00,41,00,37,00,39,00,37,00,2d,00,35,00,39,00,35,\
  00,36,00,41,00,42,00,39,00,46,00,31,00,36,00,35,00,45,00,7d,00,00,00,00,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:35,9a,3d,d6,3a,b3,a2,42,ad,ba,8b,c5,7e,21,81,11,d3,fd,b7,e3,\
  e2,1d,ce,4e,a7,97,59,56,ab,9f,16,5e

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{7F222411-5093-4B93-A63B-2996E3B2B1AC}]
"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,00,37,00,46,00,32,00,32,00,32,00,34,00,\
  31,00,31,00,2d,00,35,00,30,00,39,00,33,00,2d,00,34,00,42,00,39,00,33,00,2d,\
  00,41,00,36,00,33,00,42,00,2d,00,32,00,39,00,39,00,36,00,45,00,33,00,42,00,\
  32,00,42,00,31,00,41,00,43,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{9862DFFF-C200-43E7-A141-BC26900FD0FC}]
"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,00,39,00,38,00,36,00,32,00,44,00,46,00,\
  46,00,46,00,2d,00,43,00,32,00,30,00,30,00,2d,00,34,00,33,00,45,00,37,00,2d,\
  00,41,00,31,00,34,00,31,00,2d,00,42,00,43,00,32,00,36,00,39,00,30,00,30,00,\
  46,00,44,00,30,00,46,00,43,00,7d,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\{7F222411-5093-4B93-A63B-2996E3B2B1AC}]
"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,00,38,00,00,00,00,00
"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
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"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:46a8f3ab
"T1"=dword:46a8fab3
"T2"=dword:46a8fff9
"LeaseTerminatesTime"=dword:46a901bb
"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\{9862DFFF-C200-43E7-A141-BC26900FD0FC}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,38,00,2e,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
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"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:46a7b63e
"T1"=dword:46a7bd46
"T2"=dword:46a7c28c
"LeaseTerminatesTime"=dword:46a7c44e
"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\{D63D9A35-B33A-42A2-ADBA-8BC57E218111}]
"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\Interfaces\{E3B7FDD3-1DE2-4ECE-A797-5956AB9F165E}]
"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
```

tpccCom.tpcc_com_settingss.txt.txt

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"dlvyLogPath"="c:\\InetPub\\wwwroot\\tpcc\\"
"dlvyQueueLen"=dword:00004e20
"dlvyThreads"=dword:00000009
"nullDB"=dword:00000000
"htmlTrace"=dword:00000000
"dbName"="tpcc"
"errorLogFile"="c:\\inetpub\\wwwroot\\tpcc\\isapi_err.log"
"htmlTraceLogFile"="c:\\inetpub\\wwwroot\\tpcc\\isapi.log"
"numUsers"=dword:00007530
"dbType"="DB2"
"dbUserName"="tpcc"
"dbPassword"="tpcc"
"isapi_trace"=dword:00000000
"numServers"=dword:00000001
"numWarehouse"=dword:000009c4
"numPools"=dword:00000001
"dbInterfacePath"="C:\\inetpub\\wwwroot\\tpcc\\tpccDB2glue.dll"
```

tpcc_software_registry.reg client1

```
Transactions not supported
Enable Object pooling
Minimum pool size 42
```

Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
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 42
Maximum pool size 42
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

10.3. Linux Parameters

alloc_hugepages.sh

```
#!/bin/bash

echo  127832  > /proc/sys/vm/nr_hugepages
```

cmdline.txt

```
root=LABEL=/ selinux=0 audit=0 console=ttyS0,115200 elevator=noop splash=silent
iommu=off
```

chrt-ififo2.sh

```
#!/bin/ksh

typeset -i cnt
cnt=0
for i in `ps -eLf | grep db2sysc | awk '{print $4}'`; do
        if (($cnt == 25 )); then
                chrt -f -p 97 $i
        else
                chrt -f -p 95 $i
        fi
        cnt=($cnt+1);
done
```

zio.sh

```
#!/bin/bash
echo 500 > /sys/class/scsi_host/host0/zio_timer
echo 500 > /sys/class/scsi_host/host1/zio_timer
echo 500 > /sys/class/scsi_host/host2/zio_timer
echo 500 > /sys/class/scsi_host/host3/zio_timer
echo 500 > /sys/class/scsi_host/host4/zio_timer
echo 500 > /sys/class/scsi_host/host5/zio_timer
echo 500 > /sys/class/scsi_host/host6/zio_timer
echo 500 > /sys/class/scsi_host/host7/zio_timer
echo 500 > /sys/class/scsi_host/host8/zio_timer
echo 500 > /sys/class/scsi_host/host9/zio_timer
echo 500 > /sys/class/scsi_host/host10/zio_timer
echo 500 > /sys/class/scsi_host/host11/zio_timer
```

```
echo 1 > /sys/class/scsi_host/host0/zio
echo 1 > /sys/class/scsi_host/host1/zio
echo 1 > /sys/class/scsi_host/host2/zio
echo 1 > /sys/class/scsi_host/host3/zio
echo 1 > /sys/class/scsi_host/host4/zio
echo 1 > /sys/class/scsi_host/host5/zio
echo 1 > /sys/class/scsi_host/host6/zio
echo 1 > /sys/class/scsi_host/host7/zio
echo 1 > /sys/class/scsi_host/host8/zio
echo 1 > /sys/class/scsi_host/host9/zio
echo 1 > /sys/class/scsi_host/host10/zio
echo 1 > /sys/class/scsi_host/host11/zio
```

aff1.sh

```
#!/bin/ksh
typeset -i c
c=0
for i in `ps -eLf | grep db2sysc | awk '{print $4}'`; do
        if (($c < 6)); then
           hex=1
        fi
        if (($c == 6)); then
           hex=1
        fi
        if (($c == 7)); then
           hex=20
        fi
        if (($c == 8)); then
           hex=200
        fi
        if (($c == 9)); then
           hex=2000
        fi
        if (($c == 10)); then
           hex=2
        fi
        if (($c == 11)); then
           hex=10
        fi
        if (($c == 12)); then
           hex=100
        fi
        if (($c == 13)); then
           hex=1000
        fi
        if (($c == 14)); then
           hex=4
        fi
        if (($c == 15)); then
           hex=40
        fi
        if (($c == 16)); then
           hex=400
        fi
        if (($c == 17)); then
           hex=4000
        fi
        if (($c == 18)); then
           hex=8
        fi
        if (($c == 19)); then
           hex=80
        fi
        if (($c == 20)); then
           hex=800
        fi
        if (($c == 21)); then
           hex=8000
        fi
        cmd="taskset -p 0x${hex} $i"
        $cmd
        c=($c+1);
        if (($c > 21)); then
           break;
        fi
done
```

aff2.sh

```
#!/bin/ksh
typeset -i c
c=0
print "Cleaners:"
for i in `ps -eLf | grep db2sysc | awk '{print $4}'`; do
        hex=0
        if (($c == 28)); then
           hex=8800
        fi
        if (($c == 29)); then
           hex=8800
        fi
        if (($c == 30)); then
           hex=88
        fi
        if (($c == 31)); then
           hex=88
        fi
        if (($c == 32)); then
           hex=4400
        fi
        if (($c == 33)); then
           hex=4400
        fi
        if (($c == 34)); then
           hex=44
        fi
        if (($c == 35)); then
           hex=44
        fi
        if (($c == 36)); then
           hex=1100
        fi
        if (($c == 37)); then
           hex=1100
        fi
        if (($c == 38)); then
           hex=12
        fi
        if (($c == 39)); then
           hex=12
        fi
        if (($c == 40)); then
           hex=2200
        fi
        if (($c == 41)); then
           hex=2200
        fi
        if (($c == 42)); then
           hex=21
        fi
        if (($c == 43)); then
           hex=21
        fi
        if (($hex > 0)); then
           cmd="taskset -p 0x${hex} $i"
#          print $cmd
           $cmd
        fi
        c=($c+1);
        if (($c > 43)); then
           break;
        fi
```

aff3.sh

```
#!/bin/ksh

typeset -i cnt
cnt=0
for i in `ps -eLf | grep db2sysc | awk '{print $4}'`; do
        if (($cnt > 24 )); then
           cmd="taskset -p 0xff00 $i"
           print $cmd
           $cmd
           chrt -f -p 97 $i
           break;
        fi
        cnt=($cnt+1);
done
```

affinitize_irqs.sh

```
#!/bin/bash
echo 00000000,00000000,00000000,00000020 > /proc/irq/217/smp_affinity   # core 5
20
echo 00000000,00000000,00000000,00000200 > /proc/irq/225/smp_affinity   # core 9
200
echo 00000000,00000000,00000000,00000010 > /proc/irq/233/smp_affinity   # core 4
10
echo 00000000,00000000,00000000,00000100 > /proc/irq/50/smp_affinity    # core 8
100

echo 00000000,00000000,00000000,00000040 > /proc/irq/58/smp_affinity    # core 6   40
echo 00000000,00000000,00000000,00000400 > /proc/irq/66/smp_affinity    # core 10
400
echo 00000000,00000000,00000000,00000080 > /proc/irq/74/smp_affinity    # core 7   80
echo 00000000,00000000,00000000,00000800 > /proc/irq/82/smp_affinity    # core 4   8

# log - qlogic 2 GB port bottom chassis
echo 00000000,00000000,00000000,00008000 > /proc/irq/106/smp_affinity   # core 15
8000
echo 00000000,00000000,00000000,00008000 > /proc/irq/114/smp_affinity   # core 15
8000




/home/tpcc/affirq.sh
```

affirq.sh

```
#!/bin/ksh

# 192.168.11 network - top chassis
device=`cat /proc/interrupts | grep eth0 | awk -F : '{print $1}'`
print "eth0: echo 00000000,0000000,00000000,0000004 > /proc/irq/$device/smp_affinity"
echo "00000000,0000000,00000000,00000004" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity

device=`cat /proc/interrupts | grep eth1 | awk -F : '{print $1}'`
```

```
print "eth1: echo 00000000,0000000,00000000,0001000 > /proc/irq/$device/smp_affinity"
echo "00000000,0000000,00000000,00001000" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity
```

doit

```
# !/bin/sh

# tune slab cache
/home/tpcc/tune_slab.sh

ethtool -C eth0  tx-usecs 160 tx-frames 0 rx-usecs 300 rx-frames 0
ethtool -C eth1  tx-usecs 160 tx-frames 0 rx-usecs 300 rx-frames 0

echo "affinitizing irqs"
/home/tpcc/tpc-c.ibm/affinitize_irqs.sh

sysctl -p
/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

echo 127832 > /proc/sys/vm/nr_hugepages
```

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

```
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.sem = 500        512000  64      2048
kernel.msgmni = 4096
kernel.shmmax = 268000000000
kernel.shmall = 274877906944
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
sleep 10
raw /dev/raw/raw1       /dev/sda1
raw /dev/raw/raw2       /dev/sdb1
raw /dev/raw/raw3       /dev/sdc1
raw /dev/raw/raw4       /dev/sdd1
raw /dev/raw/raw5       /dev/sdh1
raw /dev/raw/raw6       /dev/sdg1
raw /dev/raw/raw7       /dev/sde1
raw /dev/raw/raw8       /dev/sdf1
raw /dev/raw/raw9       /dev/sdi1
raw /dev/raw/raw10      /dev/sdj1
raw /dev/raw/raw11      /dev/sdk1
raw /dev/raw/raw12      /dev/sdl1
raw /dev/raw/raw13      /dev/sdm1
raw /dev/raw/raw14      /dev/sdo1
raw /dev/raw/raw15      /dev/sdp1
raw /dev/raw/raw16      /dev/sdn1
raw /dev/raw/raw17      /dev/sdq1
raw /dev/raw/raw18      /dev/sdr1
raw /dev/raw/raw19      /dev/sds1
raw /dev/raw/raw20      /dev/sdt1
raw /dev/raw/raw21      /dev/sdu1
raw /dev/raw/raw22      /dev/sdv1
raw /dev/raw/raw23      /dev/sdw1
raw /dev/raw/raw24      /dev/sdx1
raw /dev/raw/raw25      /dev/sdz1
raw /dev/raw/raw26      /dev/sdaa1
raw /dev/raw/raw27      /dev/sdab1
raw /dev/raw/raw28      /dev/sdy1
raw /dev/raw/raw29      /dev/sdad1
raw /dev/raw/raw30      /dev/sdae1
raw /dev/raw/raw31      /dev/sdaf1
raw /dev/raw/raw32      /dev/sdac1


raw /dev/raw/raw41      /dev/sda2
raw /dev/raw/raw42      /dev/sdb2
raw /dev/raw/raw43      /dev/sdc2
raw /dev/raw/raw44      /dev/sdd2
raw /dev/raw/raw45      /dev/sdh2
raw /dev/raw/raw46      /dev/sdg2
raw /dev/raw/raw47      /dev/sde2
raw /dev/raw/raw48      /dev/sdf2
raw /dev/raw/raw49      /dev/sdi2
raw /dev/raw/raw50      /dev/sdj2
raw /dev/raw/raw51      /dev/sdk2
raw /dev/raw/raw52      /dev/sdl2
raw /dev/raw/raw53      /dev/sdm2
raw /dev/raw/raw54      /dev/sdo2
raw /dev/raw/raw55      /dev/sdp2
raw /dev/raw/raw56      /dev/sdn2
raw /dev/raw/raw57      /dev/sdq2
raw /dev/raw/raw58      /dev/sdr2
raw /dev/raw/raw59      /dev/sds2
raw /dev/raw/raw60      /dev/sdt2
raw /dev/raw/raw61      /dev/sdu2
raw /dev/raw/raw62      /dev/sdv2
raw /dev/raw/raw63      /dev/sdw2
raw /dev/raw/raw64      /dev/sdx2
raw /dev/raw/raw65      /dev/sdz2
raw /dev/raw/raw66      /dev/sdaa2
raw /dev/raw/raw67      /dev/sdab2
raw /dev/raw/raw68      /dev/sdy2
raw /dev/raw/raw69      /dev/sdad2
raw /dev/raw/raw70      /dev/sdae2
raw /dev/raw/raw71      /dev/sdaf2
raw /dev/raw/raw72      /dev/sdac2


raw /dev/raw/raw81      /dev/sda3
raw /dev/raw/raw82      /dev/sdb3
raw /dev/raw/raw83      /dev/sdc3
raw /dev/raw/raw84      /dev/sdd3
raw /dev/raw/raw85      /dev/sdh3
raw /dev/raw/raw86      /dev/sdg3
raw /dev/raw/raw87      /dev/sde3
raw /dev/raw/raw88      /dev/sdf3
raw /dev/raw/raw89      /dev/sdi3
raw /dev/raw/raw90      /dev/sdj3
raw /dev/raw/raw91      /dev/sdk3
raw /dev/raw/raw92      /dev/sdl3
raw /dev/raw/raw93      /dev/sdm3
raw /dev/raw/raw94      /dev/sdo3
raw /dev/raw/raw95      /dev/sdp3
raw /dev/raw/raw96      /dev/sdn3
raw /dev/raw/raw97      /dev/sdq3
raw /dev/raw/raw98      /dev/sdr3
raw /dev/raw/raw99      /dev/sds3
raw /dev/raw/raw100     /dev/sdt3
raw /dev/raw/raw101     /dev/sdu3
raw /dev/raw/raw102     /dev/sdv3
raw /dev/raw/raw103     /dev/sdw3
raw /dev/raw/raw104     /dev/sdx3
raw /dev/raw/raw105     /dev/sdz3
raw /dev/raw/raw106     /dev/sdaa3
raw /dev/raw/raw107     /dev/sdab3
raw /dev/raw/raw108     /dev/sdy3
raw /dev/raw/raw109     /dev/sdad3
raw /dev/raw/raw110     /dev/sdae3
raw /dev/raw/raw111     /dev/sdaf3
raw /dev/raw/raw112     /dev/sdac3

raw /dev/raw/raw121     /dev/sda5
raw /dev/raw/raw122     /dev/sdb5
raw /dev/raw/raw123     /dev/sdc5
raw /dev/raw/raw124     /dev/sdd5
raw /dev/raw/raw125     /dev/sdh5
raw /dev/raw/raw126     /dev/sdg5
raw /dev/raw/raw127     /dev/sde5
raw /dev/raw/raw128     /dev/sdf5
raw /dev/raw/raw129     /dev/sdi5
raw /dev/raw/raw130     /dev/sdj5
raw /dev/raw/raw131     /dev/sdk5
raw /dev/raw/raw132     /dev/sdl5
raw /dev/raw/raw133     /dev/sdm5
raw /dev/raw/raw134     /dev/sdo5
raw /dev/raw/raw135     /dev/sdp5
raw /dev/raw/raw136     /dev/sdn5
raw /dev/raw/raw137     /dev/sdq5
raw /dev/raw/raw138     /dev/sdr5
raw /dev/raw/raw139     /dev/sds5
raw /dev/raw/raw140     /dev/sdt5
raw /dev/raw/raw141     /dev/sdu5
raw /dev/raw/raw142     /dev/sdv5
raw /dev/raw/raw143     /dev/sdw5
raw /dev/raw/raw144     /dev/sdx5
raw /dev/raw/raw145     /dev/sdz5
raw /dev/raw/raw146     /dev/sdaa5
raw /dev/raw/raw147     /dev/sdab5
raw /dev/raw/raw148     /dev/sdy5
raw /dev/raw/raw149     /dev/sdad5
raw /dev/raw/raw150     /dev/sdae5
raw /dev/raw/raw151     /dev/sdaf5
raw /dev/raw/raw152     /dev/sdac5

raw /dev/raw/raw161     /dev/sda6
raw /dev/raw/raw162     /dev/sdb6
raw /dev/raw/raw163     /dev/sdc6
raw /dev/raw/raw164     /dev/sdd6
```

```
raw /dev/raw/raw165  /dev/sdh6
raw /dev/raw/raw166  /dev/sdg6
raw /dev/raw/raw167  /dev/sde6
raw /dev/raw/raw168  /dev/sdf6
raw /dev/raw/raw169  /dev/sdi6
raw /dev/raw/raw170  /dev/sdj6
raw /dev/raw/raw171  /dev/sdk6
raw /dev/raw/raw172  /dev/sdl6
raw /dev/raw/raw173  /dev/sdm6
raw /dev/raw/raw174  /dev/sdo6
raw /dev/raw/raw175  /dev/sdp6
raw /dev/raw/raw176  /dev/sdn6
raw /dev/raw/raw177  /dev/sdq6
raw /dev/raw/raw178  /dev/sdr6
raw /dev/raw/raw179  /dev/sds6
raw /dev/raw/raw180  /dev/sdt6
raw /dev/raw/raw181  /dev/sdu6
raw /dev/raw/raw182  /dev/sdv6
raw /dev/raw/raw183  /dev/sdw6
raw /dev/raw/raw184  /dev/sdx6
raw /dev/raw/raw185  /dev/sdz6
raw /dev/raw/raw186  /dev/sdaa6
raw /dev/raw/raw187  /dev/sdab6
raw /dev/raw/raw188  /dev/sdy6
raw /dev/raw/raw189  /dev/sdad6
raw /dev/raw/raw190  /dev/sdae6
raw /dev/raw/raw191  /dev/sdaf6
raw /dev/raw/raw192  /dev/sdac6

raw /dev/raw/raw201  /dev/sda7
raw /dev/raw/raw202  /dev/sdb7
raw /dev/raw/raw203  /dev/sdc7
raw /dev/raw/raw204  /dev/sdd7
raw /dev/raw/raw205  /dev/sdh7
raw /dev/raw/raw206  /dev/sdg7
raw /dev/raw/raw207  /dev/sde7
raw /dev/raw/raw208  /dev/sdf7
raw /dev/raw/raw209  /dev/sdi7
raw /dev/raw/raw210  /dev/sdj7
raw /dev/raw/raw211  /dev/sdk7
raw /dev/raw/raw212  /dev/sdl7
raw /dev/raw/raw213  /dev/sdm7
raw /dev/raw/raw214  /dev/sdo7
raw /dev/raw/raw215  /dev/sdp7
raw /dev/raw/raw216  /dev/sdn7
raw /dev/raw/raw217  /dev/sdq7
raw /dev/raw/raw218  /dev/sdr7
raw /dev/raw/raw219  /dev/sds7
raw /dev/raw/raw220  /dev/sdt7
raw /dev/raw/raw221  /dev/sdu7
raw /dev/raw/raw222  /dev/sdv7
raw /dev/raw/raw223  /dev/sdw7
raw /dev/raw/raw224  /dev/sdx7
raw /dev/raw/raw225  /dev/sdz7
raw /dev/raw/raw226  /dev/sdaa7
raw /dev/raw/raw227  /dev/sdab7
raw /dev/raw/raw228  /dev/sdy7
raw /dev/raw/raw229  /dev/sdad7
raw /dev/raw/raw230  /dev/sdae7
raw /dev/raw/raw231  /dev/sdaf7
raw /dev/raw/raw232  /dev/sdac7

raw /dev/raw/raw241  /dev/sda8
raw /dev/raw/raw242  /dev/sdb8
raw /dev/raw/raw243  /dev/sdc8
raw /dev/raw/raw244  /dev/sdd8
raw /dev/raw/raw245  /dev/sdh8
raw /dev/raw/raw246  /dev/sdg8
raw /dev/raw/raw247  /dev/sde8
raw /dev/raw/raw248  /dev/sdf8
raw /dev/raw/raw249  /dev/sdi8
raw /dev/raw/raw250  /dev/sdj8
raw /dev/raw/raw251  /dev/sdk8
raw /dev/raw/raw252  /dev/sdl8
raw /dev/raw/raw253  /dev/sdm8
raw /dev/raw/raw254  /dev/sdo8
raw /dev/raw/raw255  /dev/sdp8
raw /dev/raw/raw256  /dev/sdn8
raw /dev/raw/raw257  /dev/sdq8
raw /dev/raw/raw258  /dev/sdr8
raw /dev/raw/raw259  /dev/sds8
raw /dev/raw/raw260  /dev/sdt8
raw /dev/raw/raw261  /dev/sdu8
raw /dev/raw/raw262  /dev/sdv8
raw /dev/raw/raw263  /dev/sdw8
raw /dev/raw/raw264  /dev/sdx8
raw /dev/raw/raw265  /dev/sdz8
raw /dev/raw/raw266  /dev/sdaa8
raw /dev/raw/raw267  /dev/sdab8
raw /dev/raw/raw268  /dev/sdy8
raw /dev/raw/raw269  /dev/sdad8
raw /dev/raw/raw270  /dev/sdae8
raw /dev/raw/raw271  /dev/sdaf8
raw /dev/raw/raw272  /dev/sdac8

raw /dev/raw/raw281  /dev/sda9
raw /dev/raw/raw282  /dev/sdb9
raw /dev/raw/raw283  /dev/sdc9
raw /dev/raw/raw284  /dev/sdd9
raw /dev/raw/raw285  /dev/sdh9
raw /dev/raw/raw286  /dev/sdg9
raw /dev/raw/raw287  /dev/sde9
raw /dev/raw/raw288  /dev/sdf9
raw /dev/raw/raw289  /dev/sdi9
raw /dev/raw/raw290  /dev/sdj9
raw /dev/raw/raw291  /dev/sdk9
raw /dev/raw/raw292  /dev/sdl9
raw /dev/raw/raw293  /dev/sdm9
raw /dev/raw/raw294  /dev/sdo9
raw /dev/raw/raw295  /dev/sdp9
raw /dev/raw/raw296  /dev/sdn9
raw /dev/raw/raw297  /dev/sdq9
raw /dev/raw/raw298  /dev/sdr9
raw /dev/raw/raw299  /dev/sds9
raw /dev/raw/raw300  /dev/sdt9
raw /dev/raw/raw301  /dev/sdu9
raw /dev/raw/raw302  /dev/sdv9
raw /dev/raw/raw303  /dev/sdw9
raw /dev/raw/raw304  /dev/sdx9
raw /dev/raw/raw305  /dev/sdz9
raw /dev/raw/raw306  /dev/sdaa9
raw /dev/raw/raw307  /dev/sdab9
raw /dev/raw/raw308  /dev/sdy9
raw /dev/raw/raw309  /dev/sdad9
raw /dev/raw/raw310  /dev/sdae9
raw /dev/raw/raw311  /dev/sdaf9
raw /dev/raw/raw312  /dev/sdac9

raw /dev/raw/raw321  /dev/sda10
raw /dev/raw/raw322  /dev/sdb10
raw /dev/raw/raw323  /dev/sdc10
raw /dev/raw/raw324  /dev/sdd10
raw /dev/raw/raw325  /dev/sdh10
raw /dev/raw/raw326  /dev/sdg10
raw /dev/raw/raw327  /dev/sde10
raw /dev/raw/raw328  /dev/sdf10
raw /dev/raw/raw329  /dev/sdi10
raw /dev/raw/raw330  /dev/sdj10
raw /dev/raw/raw331  /dev/sdk10
raw /dev/raw/raw332  /dev/sdl10
raw /dev/raw/raw333  /dev/sdm10
raw /dev/raw/raw334  /dev/sdo10
raw /dev/raw/raw335  /dev/sdp10
raw /dev/raw/raw336  /dev/sdn10
raw /dev/raw/raw337  /dev/sdq10
raw /dev/raw/raw338  /dev/sdr10
raw /dev/raw/raw339  /dev/sds10
raw /dev/raw/raw340  /dev/sdt10
raw /dev/raw/raw341  /dev/sdu10
raw /dev/raw/raw342  /dev/sdv10
raw /dev/raw/raw343  /dev/sdw10
raw /dev/raw/raw344  /dev/sdx10
raw /dev/raw/raw345  /dev/sdz10
raw /dev/raw/raw346  /dev/sdaa10
raw /dev/raw/raw347  /dev/sdab10
raw /dev/raw/raw348  /dev/sdy10
raw /dev/raw/raw349  /dev/sdad10
raw /dev/raw/raw350  /dev/sdae10
raw /dev/raw/raw351  /dev/sdaf10
raw /dev/raw/raw352  /dev/sdac10

raw /dev/raw/raw361  /dev/sda11
raw /dev/raw/raw362  /dev/sdb11
raw /dev/raw/raw363  /dev/sdc11
raw /dev/raw/raw364  /dev/sdd11
raw /dev/raw/raw365  /dev/sdh11
raw /dev/raw/raw366  /dev/sdg11
raw /dev/raw/raw367  /dev/sde11
raw /dev/raw/raw368  /dev/sdf11
raw /dev/raw/raw369  /dev/sdi11
raw /dev/raw/raw370  /dev/sdj11
raw /dev/raw/raw371  /dev/sdk11
raw /dev/raw/raw372  /dev/sdl11
raw /dev/raw/raw373  /dev/sdm11
raw /dev/raw/raw374  /dev/sdo11
raw /dev/raw/raw375  /dev/sdp11
raw /dev/raw/raw376  /dev/sdn11
raw /dev/raw/raw377  /dev/sdq11
raw /dev/raw/raw378  /dev/sdr11
raw /dev/raw/raw379  /dev/sds11
raw /dev/raw/raw380  /dev/sdt11
raw /dev/raw/raw381  /dev/sdu11
raw /dev/raw/raw382  /dev/sdv11
raw /dev/raw/raw383  /dev/sdw11
raw /dev/raw/raw384  /dev/sdx11
raw /dev/raw/raw385  /dev/sdz11
raw /dev/raw/raw386  /dev/sdaa11
raw /dev/raw/raw387  /dev/sdab11
raw /dev/raw/raw388  /dev/sdy11
raw /dev/raw/raw389  /dev/sdad11
raw /dev/raw/raw390  /dev/sdae11
raw /dev/raw/raw391  /dev/sdaf11
raw /dev/raw/raw392  /dev/sdac11

raw /dev/raw/raw401  /dev/sda12
raw /dev/raw/raw402  /dev/sdb12
raw /dev/raw/raw403  /dev/sdc12
raw /dev/raw/raw404  /dev/sdd12
raw /dev/raw/raw405  /dev/sdh12
raw /dev/raw/raw406  /dev/sdg12
raw /dev/raw/raw407  /dev/sde12
raw /dev/raw/raw408  /dev/sdf12
raw /dev/raw/raw409  /dev/sdi12
raw /dev/raw/raw410  /dev/sdj12
raw /dev/raw/raw411  /dev/sdk12
raw /dev/raw/raw412  /dev/sdl12
raw /dev/raw/raw413  /dev/sdm12
raw /dev/raw/raw414  /dev/sdo12
raw /dev/raw/raw415  /dev/sdp12
raw /dev/raw/raw416  /dev/sdn12
```

```
raw /dev/raw/raw417     /dev/sdq12
raw /dev/raw/raw418     /dev/sdr12
raw /dev/raw/raw419     /dev/sds12
raw /dev/raw/raw420     /dev/sdt12
raw /dev/raw/raw421     /dev/sdu12
raw /dev/raw/raw422     /dev/sdv12
raw /dev/raw/raw423     /dev/sdw12
raw /dev/raw/raw424     /dev/sdx12
raw /dev/raw/raw425     /dev/sdz12
raw /dev/raw/raw426     /dev/sdaa12
raw /dev/raw/raw427     /dev/sdab12
raw /dev/raw/raw428     /dev/sdy12
raw /dev/raw/raw429     /dev/sdad12
raw /dev/raw/raw430     /dev/sdae12
raw /dev/raw/raw431     /dev/sdaf12
raw /dev/raw/raw432     /dev/sdac12

raw /dev/raw/raw441     /dev/sda13
raw /dev/raw/raw442     /dev/sdb13
raw /dev/raw/raw443     /dev/sdc13
raw /dev/raw/raw444     /dev/sdd13
raw /dev/raw/raw445     /dev/sdh13
raw /dev/raw/raw446     /dev/sdg13
raw /dev/raw/raw447     /dev/sde13
raw /dev/raw/raw448     /dev/sdf13
raw /dev/raw/raw449     /dev/sdi13
raw /dev/raw/raw450     /dev/sdj13
raw /dev/raw/raw451     /dev/sdk13
raw /dev/raw/raw452     /dev/sdl13
raw /dev/raw/raw453     /dev/sdm13
raw /dev/raw/raw454     /dev/sdo13
raw /dev/raw/raw455     /dev/sdp13
raw /dev/raw/raw456     /dev/sdn13
raw /dev/raw/raw457     /dev/sdq13
raw /dev/raw/raw458     /dev/sdr13
raw /dev/raw/raw459     /dev/sds13
raw /dev/raw/raw460     /dev/sdt13
raw /dev/raw/raw461     /dev/sdu13
raw /dev/raw/raw462     /dev/sdv13
raw /dev/raw/raw463     /dev/sdw13
raw /dev/raw/raw464     /dev/sdx13
raw /dev/raw/raw465     /dev/sdz13
raw /dev/raw/raw466     /dev/sdaa13
raw /dev/raw/raw467     /dev/sdab13
raw /dev/raw/raw468     /dev/sdy13
raw /dev/raw/raw469     /dev/sdad13
raw /dev/raw/raw470     /dev/sdae13
raw /dev/raw/raw471     /dev/sdaf13
raw /dev/raw/raw472     /dev/sdac13

raw /dev/raw/raw1300    /dev/sdag

sleep 10

chmod 777 /dev/raw/raw*
```

version.txt

```
Linux itcopus83.austin.ibm.com 2.6.18-8.el5 #1 SMP Fri Jan 26 14:15:14 EST 2007
x86_64 x86_64 x86_64 GNU/Linux
```

11 Appendix C: Database Setup Code

11.1. Database Creation Scripts

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_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 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_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_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_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_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_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_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_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_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;
connect reset;
```

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_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 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_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_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_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_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_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_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_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_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;
connect reset;
```

CRCONST_CUSTOMER10.ddl

```
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 11818 AND 13130);
SET INTEGRITY FOR CUSTOMER10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER11.ddl

```
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 13131 AND 14443);
SET INTEGRITY FOR CUSTOMER11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER12.ddl

```
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 14444 AND 15756);
SET INTEGRITY FOR CUSTOMER12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER13.ddl

```
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 15757 AND 17069);
SET INTEGRITY FOR CUSTOMER13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER14.ddl

```
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 17070 AND 18382);
SET INTEGRITY FOR CUSTOMER14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER15.ddl

```
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 18383 AND 19695);
SET INTEGRITY FOR CUSTOMER15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER16.ddl

```
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 19696 AND 21008);
SET INTEGRITY FOR CUSTOMER16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER17.ddl

```
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 21009 AND 22321);
SET INTEGRITY FOR CUSTOMER17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER18.ddl

```
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 22322 AND 23634);
SET INTEGRITY FOR CUSTOMER18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER19.ddl

```
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 23635 AND 24947);
SET INTEGRITY FOR CUSTOMER19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER1.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 1313);
SET INTEGRITY FOR CUSTOMER1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER20.ddl

```
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 24948 AND 26260);
SET INTEGRITY FOR CUSTOMER20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER21.ddl

```
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 26261 AND 27573);
SET INTEGRITY FOR CUSTOMER21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER22.ddl

```
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 27574 AND 28886);
SET INTEGRITY FOR CUSTOMER22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER23.ddl

```
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 28887 AND 30199);
SET INTEGRITY FOR CUSTOMER23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER24.ddl

```
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 30200 AND 31512);
SET INTEGRITY FOR CUSTOMER24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER25.ddl

```
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 31513 AND 32825);
SET INTEGRITY FOR CUSTOMER25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER26.ddl

```
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 32826 AND 34138);
SET INTEGRITY FOR CUSTOMER26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER27.ddl

```
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 34139 AND 35451);
SET INTEGRITY FOR CUSTOMER27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER28.ddl

```
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 35452 AND 36764);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER29.ddl

```
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 36765 AND 38077);
SET INTEGRITY FOR CUSTOMER29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER2.ddl

```
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 1314 AND 2626);
SET INTEGRITY FOR CUSTOMER2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER30.ddl

```
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 38078 AND 39390);
SET INTEGRITY FOR CUSTOMER30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER31.ddl

```
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 39391 AND 40703);
SET INTEGRITY FOR CUSTOMER31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER32.ddl

```
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
>= 40704);
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER3.ddl

```
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 2627 AND 3939);
SET INTEGRITY FOR CUSTOMER3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER4.ddl

```
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 3940 AND 5252);
SET INTEGRITY FOR CUSTOMER4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER5.ddl

```
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 5253 AND 6565);
SET INTEGRITY FOR CUSTOMER5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER6.ddl

```
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 6566 AND 7878);
SET INTEGRITY FOR CUSTOMER6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER7.ddl

```
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 7879 AND 9191);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER8.ddl

```
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 9192 AND 10504);
SET INTEGRITY FOR CUSTOMER8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_CUSTOMER9.ddl

```
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 10505 AND 11817);
SET INTEGRITY FOR CUSTOMER9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT10.ddl

```
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 11818 AND 13130);
SET INTEGRITY FOR DISTRICT10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT11.ddl

```
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 13131 AND 14443);
SET INTEGRITY FOR DISTRICT11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT12.ddl

```
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 14444 AND 15756);
SET INTEGRITY FOR DISTRICT12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT13.ddl

```
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 15757 AND 17069);
SET INTEGRITY FOR DISTRICT13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT14.ddl

```
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 17070 AND 18382);
SET INTEGRITY FOR DISTRICT14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT15.ddl

```
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 18383 AND 19695);
SET INTEGRITY FOR DISTRICT15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT16.ddl

```
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 19696 AND 21008);
SET INTEGRITY FOR DISTRICT16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT17.ddl

```
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 21009 AND 22321);
SET INTEGRITY FOR DISTRICT17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT18.ddl

```
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 22322 AND 23634);
SET INTEGRITY FOR DISTRICT18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT19.ddl

```
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 23635 AND 24947);
SET INTEGRITY FOR DISTRICT19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT1.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 1313);
SET INTEGRITY FOR DISTRICT1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT20.ddl

```
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 24948 AND 26260);
SET INTEGRITY FOR DISTRICT20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT21.ddl

```
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 26261 AND 27573);
SET INTEGRITY FOR DISTRICT21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT22.ddl

```
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 27574 AND 28886);
SET INTEGRITY FOR DISTRICT22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT23.ddl

```
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 28887 AND 30199);
SET INTEGRITY FOR DISTRICT23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT24.ddl

```
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 30200 AND 31512);
SET INTEGRITY FOR DISTRICT24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT25.ddl

```
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 31513 AND 32825);
SET INTEGRITY FOR DISTRICT25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT26.ddl

```
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 32826 AND 34138);
SET INTEGRITY FOR DISTRICT26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT27.ddl

```
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 34139 AND 35451);
SET INTEGRITY FOR DISTRICT27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT28.ddl

```
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 35452 AND 36764);
SET INTEGRITY FOR DISTRICT28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT29.ddl

```
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 36765 AND 38077);
SET INTEGRITY FOR DISTRICT29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT2.ddl

```
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 1314 AND 2626);
SET INTEGRITY FOR DISTRICT2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT30.ddl

```
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 38078 AND 39390);
SET INTEGRITY FOR DISTRICT30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT31.ddl

```
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 39391 AND 40703);
SET INTEGRITY FOR DISTRICT31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT32.ddl

```
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
>= 40704);
SET INTEGRITY FOR DISTRICT32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT3.ddl

```
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 2627 AND 3939);
SET INTEGRITY FOR DISTRICT3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT4.ddl

```
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 3940 AND 5252);
SET INTEGRITY FOR DISTRICT4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT5.ddl

```
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 5253 AND 6565);
SET INTEGRITY FOR DISTRICT5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT6.ddl

```
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 6566 AND 7878);
SET INTEGRITY FOR DISTRICT6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT7.ddl

```
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 7879 AND 9191);
SET INTEGRITY FOR DISTRICT7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT8.ddl

```
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 9192 AND 10504);
SET INTEGRITY FOR DISTRICT8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_DISTRICT9.ddl

```
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 10505 AND 11817);
SET INTEGRITY FOR DISTRICT9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY10.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY10 OFF;
ALTER TABLE HISTORY10 DROP CONSTRAINT HISTORY10CKC;
ALTER TABLE HISTORY10 ADD CONSTRAINT HISTORY10CKC CHECK (H_W_ID
BETWEEN 11818 AND 13130);
SET INTEGRITY FOR HISTORY10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY11.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY11 OFF;
ALTER TABLE HISTORY11 DROP CONSTRAINT HISTORY11CKC;
ALTER TABLE HISTORY11 ADD CONSTRAINT HISTORY11CKC CHECK (H_W_ID
BETWEEN 13131 AND 14443);
SET INTEGRITY FOR HISTORY11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY12.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY12 OFF;
ALTER TABLE HISTORY12 DROP CONSTRAINT HISTORY12CKC;
ALTER TABLE HISTORY12 ADD CONSTRAINT HISTORY12CKC CHECK (H_W_ID
BETWEEN 14444 AND 15756);
SET INTEGRITY FOR HISTORY12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY13.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY13 OFF;
ALTER TABLE HISTORY13 DROP CONSTRAINT HISTORY13CKC;
ALTER TABLE HISTORY13 ADD CONSTRAINT HISTORY13CKC CHECK (H_W_ID
BETWEEN 15757 AND 17069);
SET INTEGRITY FOR HISTORY13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY14.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY14 OFF;
ALTER TABLE HISTORY14 DROP CONSTRAINT HISTORY14CKC;
ALTER TABLE HISTORY14 ADD CONSTRAINT HISTORY14CKC CHECK (H_W_ID
BETWEEN 17070 AND 18382);
SET INTEGRITY FOR HISTORY14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY15.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY15 OFF;
ALTER TABLE HISTORY15 DROP CONSTRAINT HISTORY15CKC;
ALTER TABLE HISTORY15 ADD CONSTRAINT HISTORY15CKC CHECK (H_W_ID
BETWEEN 18383 AND 19695);
SET INTEGRITY FOR HISTORY15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY16.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY16 OFF;
ALTER TABLE HISTORY16 DROP CONSTRAINT HISTORY16CKC;
ALTER TABLE HISTORY16 ADD CONSTRAINT HISTORY16CKC CHECK (H_W_ID
BETWEEN 19696 AND 21008);
SET INTEGRITY FOR HISTORY16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY17.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY17 OFF;
ALTER TABLE HISTORY17 DROP CONSTRAINT HISTORY17CKC;
ALTER TABLE HISTORY17 ADD CONSTRAINT HISTORY17CKC CHECK (H_W_ID
BETWEEN 21009 AND 22321);
SET INTEGRITY FOR HISTORY17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY18.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY18 OFF;
ALTER TABLE HISTORY18 DROP CONSTRAINT HISTORY18CKC;
ALTER TABLE HISTORY18 ADD CONSTRAINT HISTORY18CKC CHECK (H_W_ID
BETWEEN 22322 AND 23634);
SET INTEGRITY FOR HISTORY18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY19.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY19 OFF;
ALTER TABLE HISTORY19 DROP CONSTRAINT HISTORY19CKC;
ALTER TABLE HISTORY19 ADD CONSTRAINT HISTORY19CKC CHECK (H_W_ID
BETWEEN 23635 AND 24947);
SET INTEGRITY FOR HISTORY19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY1.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY1 OFF;
ALTER TABLE HISTORY1 DROP CONSTRAINT HISTORY1CKC;
ALTER TABLE HISTORY1 ADD CONSTRAINT HISTORY1CKC CHECK (H_W_ID
BETWEEN 1 AND 1313);
SET INTEGRITY FOR HISTORY1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY20.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY20 OFF;
ALTER TABLE HISTORY20 DROP CONSTRAINT HISTORY20CKC;
ALTER TABLE HISTORY20 ADD CONSTRAINT HISTORY20CKC CHECK (H_W_ID
BETWEEN 24948 AND 26260);
SET INTEGRITY FOR HISTORY20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY21.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY21 OFF;
ALTER TABLE HISTORY21 DROP CONSTRAINT HISTORY21CKC;
ALTER TABLE HISTORY21 ADD CONSTRAINT HISTORY21CKC CHECK (H_W_ID
BETWEEN 26261 AND 27573);
SET INTEGRITY FOR HISTORY21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY22.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY22 OFF;
ALTER TABLE HISTORY22 DROP CONSTRAINT HISTORY22CKC;
ALTER TABLE HISTORY22 ADD CONSTRAINT HISTORY22CKC CHECK (H_W_ID
BETWEEN 27574 AND 28886);
SET INTEGRITY FOR HISTORY22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY23.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY23 OFF;
ALTER TABLE HISTORY23 DROP CONSTRAINT HISTORY23CKC;
ALTER TABLE HISTORY23 ADD CONSTRAINT HISTORY23CKC CHECK (H_W_ID
BETWEEN 28887 AND 30199);
SET INTEGRITY FOR HISTORY23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY24.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY24 OFF;
ALTER TABLE HISTORY24 DROP CONSTRAINT HISTORY24CKC;
ALTER TABLE HISTORY24 ADD CONSTRAINT HISTORY24CKC CHECK (H_W_ID
BETWEEN 30200 AND 31512);
SET INTEGRITY FOR HISTORY24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY25.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY25 OFF;
ALTER TABLE HISTORY25 DROP CONSTRAINT HISTORY25CKC;
ALTER TABLE HISTORY25 ADD CONSTRAINT HISTORY25CKC CHECK (H_W_ID
BETWEEN 31513 AND 32825);
SET INTEGRITY FOR HISTORY25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY26.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY26 OFF;
ALTER TABLE HISTORY26 DROP CONSTRAINT HISTORY26CKC;
ALTER TABLE HISTORY26 ADD CONSTRAINT HISTORY26CKC CHECK (H_W_ID
BETWEEN 32826 AND 34138);
SET INTEGRITY FOR HISTORY26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY27.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY27 OFF;
ALTER TABLE HISTORY27 DROP CONSTRAINT HISTORY27CKC;
ALTER TABLE HISTORY27 ADD CONSTRAINT HISTORY27CKC CHECK (H_W_ID
BETWEEN 34139 AND 35451);
SET INTEGRITY FOR HISTORY27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY28.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY28 OFF;
ALTER TABLE HISTORY28 DROP CONSTRAINT HISTORY28CKC;
ALTER TABLE HISTORY28 ADD CONSTRAINT HISTORY28CKC CHECK (H_W_ID
BETWEEN 35452 AND 36764);
SET INTEGRITY FOR HISTORY28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY29.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY29 OFF;
ALTER TABLE HISTORY29 DROP CONSTRAINT HISTORY29CKC;
ALTER TABLE HISTORY29 ADD CONSTRAINT HISTORY29CKC CHECK (H_W_ID
BETWEEN 36765 AND 38077);
SET INTEGRITY FOR HISTORY29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY2.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY2 OFF;
ALTER TABLE HISTORY2 DROP CONSTRAINT HISTORY2CKC;
ALTER TABLE HISTORY2 ADD CONSTRAINT HISTORY2CKC CHECK (H_W_ID
BETWEEN 1314 AND 2626);
SET INTEGRITY FOR HISTORY2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY30.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY30 OFF;
ALTER TABLE HISTORY30 DROP CONSTRAINT HISTORY30CKC;
ALTER TABLE HISTORY30 ADD CONSTRAINT HISTORY30CKC CHECK (H_W_ID
BETWEEN 38078 AND 39390);
SET INTEGRITY FOR HISTORY30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY31.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY31 OFF;
ALTER TABLE HISTORY31 DROP CONSTRAINT HISTORY31CKC;
ALTER TABLE HISTORY31 ADD CONSTRAINT HISTORY31CKC CHECK (H_W_ID
BETWEEN 39391 AND 40703);
SET INTEGRITY FOR HISTORY31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY32.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY32 OFF;
ALTER TABLE HISTORY32 DROP CONSTRAINT HISTORY32CKC;
ALTER TABLE HISTORY32 ADD CONSTRAINT HISTORY32CKC CHECK (H_W_ID >=
40704);
SET INTEGRITY FOR HISTORY32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY3.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY3 OFF;
ALTER TABLE HISTORY3 DROP CONSTRAINT HISTORY3CKC;
ALTER TABLE HISTORY3 ADD CONSTRAINT HISTORY3CKC CHECK (H_W_ID
BETWEEN 2627 AND 3939);
SET INTEGRITY FOR HISTORY3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY4.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY4 OFF;
ALTER TABLE HISTORY4 DROP CONSTRAINT HISTORY4CKC;
ALTER TABLE HISTORY4 ADD CONSTRAINT HISTORY4CKC CHECK (H_W_ID
BETWEEN 3940 AND 5252);
SET INTEGRITY FOR HISTORY4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY5.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY5 OFF;
ALTER TABLE HISTORY5 DROP CONSTRAINT HISTORY5CKC;
ALTER TABLE HISTORY5 ADD CONSTRAINT HISTORY5CKC CHECK (H_W_ID
BETWEEN 5253 AND 6565);
SET INTEGRITY FOR HISTORY5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY6.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY6 OFF;
ALTER TABLE HISTORY6 DROP CONSTRAINT HISTORY6CKC;
ALTER TABLE HISTORY6 ADD CONSTRAINT HISTORY6CKC CHECK (H_W_ID
BETWEEN 6566 AND 7878);
SET INTEGRITY FOR HISTORY6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY7.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY7 OFF;
ALTER TABLE HISTORY7 DROP CONSTRAINT HISTORY7CKC;
ALTER TABLE HISTORY7 ADD CONSTRAINT HISTORY7CKC CHECK (H_W_ID
BETWEEN 7879 AND 9191);
SET INTEGRITY FOR HISTORY7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY8.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY8 OFF;
ALTER TABLE HISTORY8 DROP CONSTRAINT HISTORY8CKC;
ALTER TABLE HISTORY8 ADD CONSTRAINT HISTORY8CKC CHECK (H_W_ID
BETWEEN 9192 AND 10504);
SET INTEGRITY FOR HISTORY8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_HISTORY9.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY9 OFF;
ALTER TABLE HISTORY9 DROP CONSTRAINT HISTORY9CKC;
ALTER TABLE HISTORY9 ADD CONSTRAINT HISTORY9CKC CHECK (H_W_ID
BETWEEN 10505 AND 11817);
SET INTEGRITY FOR HISTORY9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA10.ddl

```
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 11818 AND 13130) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA11.ddl

```
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 13131 AND 14443) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA12.ddl

```
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 14444 AND 15756) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA13.ddl

```
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 15757 AND 17069) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA14.ddl

```
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 17070 AND 18382) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA15.ddl

```
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 18383 AND 19695) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA16.ddl

```
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 19696 AND 21008) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA17.ddl

```
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 21009 AND 22321) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA18.ddl

```
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 22322 AND 23634) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA19.ddl

```
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 23635 AND 24947) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA1.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 1313) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA20.ddl

```
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 24948 AND 26260) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA21.ddl

```
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 26261 AND 27573) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA22.ddl

```
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 27574 AND 28886) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA23.ddl

```
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 28887 AND 30199) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA24.ddl

```
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 30200 AND 31512) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA25.ddl

```
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 31513 AND 32825) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA26.ddl

```
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 32826 AND 34138) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA27.ddl

```
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 34139 AND 35451) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA28.ddl

```
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 35452 AND 36764) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA29.ddl

```
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 36765 AND 38077) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA2.ddl

```
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 1314 AND 2626) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA30.ddl

```
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 38078 AND 39390) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA31.ddl

```
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 39391 AND 40703) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA32.ddl

```
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 >= 40704) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA3.ddl

```
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 2627 AND 3939) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA4.ddl

```
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 3940 AND 5252) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA5.ddl

```
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 5253 AND 6565) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA6.ddl

```
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 6566 AND 7878) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA7.ddl

```
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 7879 AND 9191) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA8.ddl

```
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 9192 AND 10504) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERA9.ddl

```
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 10505 AND 11817) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB10.ddl

```
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 11818 AND 13130) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB11.ddl

```
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 13131 AND 14443) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB12.ddl

```
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 14444 AND 15756) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB13.ddl

```
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 15757 AND 17069) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB14.ddl

```
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 17070 AND 18382) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB15.ddl

```
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 18383 AND 19695) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB16.ddl

```
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 19696 AND 21008) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB17.ddl

```
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 21009 AND 22321) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB18.ddl

```
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 22322 AND 23634) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB19.ddl

```
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 23635 AND 24947) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB1.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 1313) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB20.ddl

```
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 24948 AND 26260) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB21.ddl

```
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 26261 AND 27573) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB22.ddl

```
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 27574 AND 28886) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB23.ddl

```
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 28887 AND 30199) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB24.ddl

```
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 30200 AND 31512) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB25.ddl

```
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 31513 AND 32825) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB26.ddl

```
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 32826 AND 34138) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB27.ddl

```
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 34139 AND 35451) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB28.ddl

```
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 35452 AND 36764) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB29.ddl

```
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 36765 AND 38077) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB2.ddl

```
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 1314 AND 2626) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB30.ddl

```
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 38078 AND 39390) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB31.ddl

```
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 39391 AND 40703) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB32.ddl

```
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 >= 40704) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB3.ddl

```
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 2627 AND 3939) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB4.ddl

```
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 3940 AND 5252) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB5.ddl

```
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 5253 AND 6565) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB6.ddl

```
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 6566 AND 7878) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB7.ddl

```
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 7879 AND 9191) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB8.ddl

```
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 9192 AND 10504) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_NEW_ORDERB9.ddl

```
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 10505 AND 11817) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE10.ddl

```
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 11818 AND 13130);
SET INTEGRITY FOR ORDER_LINE10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE11.ddl

```
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 13131 AND 14443);
SET INTEGRITY FOR ORDER_LINE11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE12.ddl

```
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 14444 AND 15756);
SET INTEGRITY FOR ORDER_LINE12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE13.ddl

```
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 15757 AND 17069);
SET INTEGRITY FOR ORDER_LINE13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE14.ddl

```
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 17070 AND 18382);
SET INTEGRITY FOR ORDER_LINE14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE15.ddl

```
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 18383 AND 19695);
SET INTEGRITY FOR ORDER_LINE15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE16.ddl

```
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 19696 AND 21008);
SET INTEGRITY FOR ORDER_LINE16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE17.ddl

```
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 21009 AND 22321);
SET INTEGRITY FOR ORDER_LINE17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE18.ddl

```
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 22322 AND 23634);
SET INTEGRITY FOR ORDER_LINE18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE19.ddl

```
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 23635 AND 24947);
SET INTEGRITY FOR ORDER_LINE19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE1.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 1313);
SET INTEGRITY FOR ORDER_LINE1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE20.ddl

```
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 24948 AND 26260);
SET INTEGRITY FOR ORDER_LINE20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE21.ddl

```
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 26261 AND 27573);
SET INTEGRITY FOR ORDER_LINE21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE22.ddl

```
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 27574 AND 28886);
SET INTEGRITY FOR ORDER_LINE22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE23.ddl

```
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 28887 AND 30199);
SET INTEGRITY FOR ORDER_LINE23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE24.ddl

```
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 30200 AND 31512);
SET INTEGRITY FOR ORDER_LINE24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE25.ddl

```
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 31513 AND 32825);
SET INTEGRITY FOR ORDER_LINE25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE26.ddl

```
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 32826 AND 34138);
SET INTEGRITY FOR ORDER_LINE26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE27.ddl

```
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 34139 AND 35451);
SET INTEGRITY FOR ORDER_LINE27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE28.ddl

```
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 35452 AND 36764);
SET INTEGRITY FOR ORDER_LINE28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE29.ddl

```
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 36765 AND 38077);
SET INTEGRITY FOR ORDER_LINE29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE2.ddl

```
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 1314 AND 2626);
SET INTEGRITY FOR ORDER_LINE2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE30.ddl

```
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 38078 AND 39390);
SET INTEGRITY FOR ORDER_LINE30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE31.ddl

```
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 39391 AND 40703);
SET INTEGRITY FOR ORDER_LINE31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE32.ddl

```
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 >= 40704);
SET INTEGRITY FOR ORDER_LINE32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE3.ddl

```
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 2627 AND 3939);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE4.ddl

```
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 3940 AND 5252);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE5.ddl

```
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 5253 AND 6565);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE6.ddl

```
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 6566 AND 7878);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE7.ddl

```
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 7879 AND 9191);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE8.ddl

```
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 9192 AND 10504);
SET INTEGRITY FOR ORDER_LINE8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDER_LINE9.ddl

```
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 10505 AND 11817);
SET INTEGRITY FOR ORDER_LINE9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS10.ddl

```
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 11818 AND 13130);
SET INTEGRITY FOR ORDERS10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS11.ddl

```
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 13131 AND 14443);
SET INTEGRITY FOR ORDERS11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS12.ddl

```
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 14444 AND 15756);
SET INTEGRITY FOR ORDERS12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS13.ddl

```
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 15757 AND 17069);
SET INTEGRITY FOR ORDERS13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS14.ddl

```
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 17070 AND 18382);
SET INTEGRITY FOR ORDERS14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS15.ddl

```
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 18383 AND 19695);
SET INTEGRITY FOR ORDERS15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS16.ddl

```
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 19696 AND 21008);
SET INTEGRITY FOR ORDERS16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS17.ddl

```
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 21009 AND 22321);
SET INTEGRITY FOR ORDERS17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS18.ddl

```
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 22322 AND 23634);
SET INTEGRITY FOR ORDERS18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS19.ddl

```
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 23635 AND 24947);
SET INTEGRITY FOR ORDERS19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS1.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 1313);
SET INTEGRITY FOR ORDERS1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS20.ddl

```
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 24948 AND 26260);
SET INTEGRITY FOR ORDERS20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS21.ddl

```
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 26261 AND 27573);
SET INTEGRITY FOR ORDERS21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS22.ddl

```
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 27574 AND 28886);
SET INTEGRITY FOR ORDERS22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS23.ddl

```
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 28887 AND 30199);
SET INTEGRITY FOR ORDERS23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS24.ddl

```
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 30200 AND 31512);
SET INTEGRITY FOR ORDERS24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS25.ddl

```
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 31513 AND 32825);
SET INTEGRITY FOR ORDERS25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS26.ddl

```
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 32826 AND 34138);
SET INTEGRITY FOR ORDERS26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS27.ddl

```
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 34139 AND 35451);
SET INTEGRITY FOR ORDERS27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS28.ddl

```
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 35452 AND 36764);
SET INTEGRITY FOR ORDERS28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS29.ddl

```
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 36765 AND 38077);
SET INTEGRITY FOR ORDERS29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS2.ddl

```
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 1314 AND 2626);
SET INTEGRITY FOR ORDERS2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS30.ddl

```
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 38078 AND 39390);
SET INTEGRITY FOR ORDERS30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS31.ddl

```
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 39391 AND 40703);
SET INTEGRITY FOR ORDERS31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS32.ddl

```
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 >=
40704);
SET INTEGRITY FOR ORDERS32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS3.ddl

```
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 2627 AND 3939);
SET INTEGRITY FOR ORDERS3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS4.ddl

```
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 3940 AND 5252);
SET INTEGRITY FOR ORDERS4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS5.ddl

```
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 5253 AND 6565);
SET INTEGRITY FOR ORDERS5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS6.ddl

```
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 6566 AND 7878);
SET INTEGRITY FOR ORDERS6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS7.ddl

```
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 7879 AND 9191);
SET INTEGRITY FOR ORDERS7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS8.ddl

```
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 9192 AND 10504);
SET INTEGRITY FOR ORDERS8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_ORDERS9.ddl

```
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 10505 AND 11817);
SET INTEGRITY FOR ORDERS9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK10.ddl

```
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 11818 AND 13130);
SET INTEGRITY FOR STOCK10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK11.ddl

```
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 13131 AND 14443);
SET INTEGRITY FOR STOCK11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK12.ddl

```
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 14444 AND 15756);
SET INTEGRITY FOR STOCK12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK13.ddl

```
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 15757 AND 17069);
SET INTEGRITY FOR STOCK13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK14.ddl

```
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 17070 AND 18382);
SET INTEGRITY FOR STOCK14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK15.ddl

```
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 18383 AND 19695);
SET INTEGRITY FOR STOCK15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK16.ddl

```
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 19696 AND 21008);
SET INTEGRITY FOR STOCK16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK17.ddl

```
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 21009 AND 22321);
SET INTEGRITY FOR STOCK17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK18.ddl

```
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 22322 AND 23634);
SET INTEGRITY FOR STOCK18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK19.ddl

```
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 23635 AND 24947);
SET INTEGRITY FOR STOCK19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK1.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 1313);
SET INTEGRITY FOR STOCK1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK20.ddl

```
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 24948 AND 26260);
SET INTEGRITY FOR STOCK20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK21.ddl

```
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 26261 AND 27573);
SET INTEGRITY FOR STOCK21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK22.ddl

```
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 27574 AND 28886);
SET INTEGRITY FOR STOCK22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK23.ddl

```
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 28887 AND 30199);
SET INTEGRITY FOR STOCK23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK24.ddl

```
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 30200 AND 31512);
SET INTEGRITY FOR STOCK24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK25.ddl

```
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 31513 AND 32825);
SET INTEGRITY FOR STOCK25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK26.ddl

```
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 32826 AND 34138);
SET INTEGRITY FOR STOCK26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK27.ddl

```
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 34139 AND 35451);
SET INTEGRITY FOR STOCK27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK28.ddl

```
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 35452 AND 36764);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK29.ddl

```
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 36765 AND 38077);
SET INTEGRITY FOR STOCK29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK2.ddl

```
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
1314 AND 2626);
SET INTEGRITY FOR STOCK2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK30.ddl

```
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 38078 AND 39390);
SET INTEGRITY FOR STOCK30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK31.ddl

```
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 39391 AND 40703);
SET INTEGRITY FOR STOCK31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK32.ddl

```
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 >=
40704);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK3.ddl

```
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
2627 AND 3939);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK4.ddl

```
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
3940 AND 5252);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK5.ddl

```
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
5253 AND 6565);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK6.ddl

```
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
6566 AND 7878);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK7.ddl

```
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
7879 AND 9191);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK8.ddl

```
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
9192 AND 10504);
SET INTEGRITY FOR STOCK8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_STOCK9.ddl

```
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
10505 AND 11817);
SET INTEGRITY FOR STOCK9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE10.ddl

```
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 11818 AND 13130);
SET INTEGRITY FOR WAREHOUSE10 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE11.ddl

```
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 13131 AND 14443);
SET INTEGRITY FOR WAREHOUSE11 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE12.ddl

```
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 14444 AND 15756);
SET INTEGRITY FOR WAREHOUSE12 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE13.ddl

```
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 15757 AND 17069);
SET INTEGRITY FOR WAREHOUSE13 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE14.ddl

```
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 17070 AND 18382);
SET INTEGRITY FOR WAREHOUSE14 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE15.ddl

```
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 18383 AND 19695);
SET INTEGRITY FOR WAREHOUSE15 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE16.ddl

```
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 19696 AND 21008);
SET INTEGRITY FOR WAREHOUSE16 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE17.ddl

```
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 21009 AND 22321);
SET INTEGRITY FOR WAREHOUSE17 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE18.ddl

```
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 22322 AND 23634);
SET INTEGRITY FOR WAREHOUSE18 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE19.ddl

```
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 23635 AND 24947);
SET INTEGRITY FOR WAREHOUSE19 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE1.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 1313);
SET INTEGRITY FOR WAREHOUSE1 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE20.ddl

```
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 24948 AND 26260);
SET INTEGRITY FOR WAREHOUSE20 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE21.ddl

```
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 26261 AND 27573);
SET INTEGRITY FOR WAREHOUSE21 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE22.ddl

```
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 27574 AND 28886);
SET INTEGRITY FOR WAREHOUSE22 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE23.ddl

```
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 28887 AND 30199);
SET INTEGRITY FOR WAREHOUSE23 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE24.ddl

```
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 30200 AND 31512);
SET INTEGRITY FOR WAREHOUSE24 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE25.ddl

```
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 31513 AND 32825);
SET INTEGRITY FOR WAREHOUSE25 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE26.ddl

```
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 32826 AND 34138);
SET INTEGRITY FOR WAREHOUSE26 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE27.ddl

```
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 34139 AND 35451);
SET INTEGRITY FOR WAREHOUSE27 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE28.ddl

```
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 35452 AND 36764);
SET INTEGRITY FOR WAREHOUSE28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE29.ddl

```
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 36765 AND 38077);
SET INTEGRITY FOR WAREHOUSE29 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE2.ddl

```
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 1314 AND 2626);
SET INTEGRITY FOR WAREHOUSE2 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE30.ddl

```
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 38078 AND 39390);
SET INTEGRITY FOR WAREHOUSE30 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE31.ddl

```
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 39391 AND 40703);
SET INTEGRITY FOR WAREHOUSE31 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE32.ddl

```
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
>= 40704);
SET INTEGRITY FOR WAREHOUSE32 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE3.ddl

```
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 2627 AND 3939);
SET INTEGRITY FOR WAREHOUSE3 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE4.ddl

```
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 3940 AND 5252);
SET INTEGRITY FOR WAREHOUSE4 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE5.ddl

```
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 5253 AND 6565);
SET INTEGRITY FOR WAREHOUSE5 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE6.ddl

```
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 6566 AND 7878);
SET INTEGRITY FOR WAREHOUSE6 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE7.ddl

```
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 7879 AND 9191);
SET INTEGRITY FOR WAREHOUSE7 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE8.ddl

```
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 9192 AND 10504);
SET INTEGRITY FOR WAREHOUSE8 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRCONST_WAREHOUSE9.ddl

```
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 10505 AND 11817);
SET INTEGRITY FOR WAREHOUSE9 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRIDX_CUST_IDXB10.ddl

```
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;
```

CRIDX_CUST_IDXB11.ddl

```
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;
```

CRIDX_CUST_IDXB12.ddl

```
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;
```

CRIDX_CUST_IDXB13.ddl

```
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;
```

CRIDX_CUST_IDXB14.ddl

```
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;
```

CRIDX_CUST_IDXB15.ddl

```
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;
```

CRIDX_CUST_IDXB16.ddl

```
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;
```

CRIDX_CUST_IDXB17.ddl

```
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;
```

CRIDX_CUST_IDXB18.ddl

```
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;
```

CRIDX_CUST_IDXB19.ddl

```
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;
```

CRIDX_CUST_IDXB1.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;
```

CRIDX_CUST_IDXB20.ddl

```
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;
```

CRIDX_CUST_IDXB21.ddl

```
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;
```

CRIDX_CUST_IDXB22.ddl

```
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;
```

CRIDX_CUST_IDXB23.ddl

```
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;
```

CRIDX_CUST_IDXB24.ddl

```
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;
```

CRIDX_CUST_IDXB25.ddl

```
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;
```

CRIDX_CUST_IDXB26.ddl

```
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;
```

CRIDX_CUST_IDXB27.ddl

```
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;
```

CRIDX_CUST_IDXB28.ddl

```
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;
```

CRIDX_CUST_IDXB29.ddl

```
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;
```

CRIDX_CUST_IDXB2.ddl

```
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;
```

CRIDX_CUST_IDXB30.ddl

```
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;
```

CRIDX_CUST_IDXB31.ddl

```
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;
```

CRIDX_CUST_IDXB32.ddl

```
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;
```

CRIDX_CUST_IDXB3.ddl

```
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;
```

CRIDX_CUST_IDXB4.ddl

```
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;
```

CRIDX_CUST_IDXB5.ddl

```
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;
```

CRIDX_CUST_IDXB6.ddl

```
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;
```

CRIDX_CUST_IDXB7.ddl

```
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;
```

CRIDX_CUST_IDXB8.ddl

```
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;
```

CRIDX_CUST_IDXB9.ddl

```
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;
```

CRIDX_ORDR_IDXB10.ddl

```
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;
```

CRIDX_ORDR_IDXB11.ddl

```
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;
```

CRIDX_ORDR_IDXB12.ddl

```
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;
```

CRIDX_ORDR_IDXB13.ddl

```
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;
```

CRIDX_ORDR_IDXB14.ddl

```
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;
```

CRIDX_ORDR_IDXB15.ddl

```
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;
```

CRIDX_ORDR_IDXB16.ddl

```
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;
```

CRIDX_ORDR_IDXB17.ddl

```
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;
```

CRIDX_ORDR_IDXB18.ddl

```
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;
```

CRIDX_ORDR_IDXB19.ddl

```
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;
```

CRIDX_ORDR_IDXB1.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;
```

CRIDX_ORDR_IDXB20.ddl

```
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;
```

CRIDX_ORDR_IDXB21.ddl

```
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;
```

CRIDX_ORDR_IDXB22.ddl

```
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;
```

CRIDX_ORDR_IDXB23.ddl

```
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;
```

CRIDX_ORDR_IDXB24.ddl

```
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;
```

CRIDX_ORDR_IDXB25.ddl

```
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;
```

CRIDX_ORDR_IDXB26.ddl

```
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;
```

CRIDX_ORDR_IDXB27.ddl

```
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;
```

CRIDX_ORDR_IDXB28.ddl

```
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;
```

CRIDX_ORDR_IDXB29.ddl

```
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;
```

CRIDX_ORDR_IDXB2.ddl

```
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;
```

CRIDX_ORDR_IDXB30.ddl

```
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;
```

CRIDX_ORDR_IDXB31.ddl

```
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;
```

CRIDX_ORDR_IDXB32.ddl

```
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;
```

CRIDX_ORDR_IDXB3.ddl

```
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;
```

CRIDX_ORDR_IDXB4.ddl

```
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;
```

CRIDX_ORDR_IDXB5.ddl

```
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;
```

CRIDX_ORDR_IDXB6.ddl

```
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;
```

CRIDX_ORDR_IDXB7.ddl

```
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;
```

CRIDX_ORDR_IDXB8.ddl

```
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;
```

CRIDX_ORDR_IDXB9.ddl

```
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;
```

CRTB_CUSTOMER10.ddl

```
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 11818 ENDING AT 13130,
                     C_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_CUSTOMER11.ddl

```
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 13131 ENDING AT 14443,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER12.ddl

```
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 14444 ENDING AT 15756,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER13.ddl

```
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 15757 ENDING AT 17069,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER14.ddl

```
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 17070 ENDING AT 18382,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER15.ddl

```
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 18383 ENDING AT 19695,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER16.ddl

```
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 19696 ENDING AT 21008,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER17.ddl

```
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 21009 ENDING AT 22321,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER18.ddl

```
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 22322 ENDING AT 23634,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER19.ddl

```
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 23635 ENDING AT 24947,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER1.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 1313,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER20.ddl

```
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 24948 ENDING AT 26260,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER21.ddl

```
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 26261 ENDING AT 27573,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER22.ddl

```
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 27574 ENDING AT 28886,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER23.ddl

```
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 28887 ENDING AT 30199,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER24.ddl

```
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 30200 ENDING AT 31512,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER25.ddl

```
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 31513 ENDING AT 32825,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER26.ddl

```
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 32826 ENDING AT 34138,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER27.ddl

```
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 34139 ENDING AT 35451,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER28.ddl

```
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 35452 ENDING AT 36764,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER29.ddl

```
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 36765 ENDING AT 38077,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER2.ddl

```
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 1314 ENDING AT 2626,
                   C_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER30.ddl

```
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 38078 ENDING AT 39390,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER31.ddl

```
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 39391 ENDING AT 40703,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER32.ddl

```
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 40704 ENDING AT 42016,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER3.ddl

```
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 2627 ENDING AT 3939,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER4.ddl

```
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 3940 ENDING AT 5252,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER5.ddl

```
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 5253 ENDING AT 6565,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER6.ddl

```
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 6566 ENDING AT 7878,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER7.ddl

```
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 7879 ENDING AT 9191,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER8.ddl

```
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 9192 ENDING AT 10504,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER9.ddl

```
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 10505 ENDING AT 11817,
                 C_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT10.ddl

```
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 11818 ENDING AT 13130
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT11.ddl

```
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 13131 ENDING AT 14443
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT12.ddl

```
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 14444 ENDING AT 15756
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT13.ddl

```
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 15757 ENDING AT 17069
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT14.ddl

```
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 17070 ENDING AT 18382
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT15.ddl

```
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 18383 ENDING AT 19695
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT16.ddl

```
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 19696 ENDING AT 21008
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT17.ddl

```
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 21009 ENDING AT 22321
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT18.ddl

```
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 22322 ENDING AT 23634
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT19.ddl

```
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 23635 ENDING AT 24947
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT1.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 1313
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT20.ddl

```
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 24948 ENDING AT 26260
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT21.ddl

```
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 26261 ENDING AT 27573
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT22.ddl

```
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 27574 ENDING AT 28886
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT23.ddl

```
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 28887 ENDING AT 30199
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT24.ddl

```
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 30200 ENDING AT 31512
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT25.ddl

```
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 31513 ENDING AT 32825
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT26.ddl

```
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 32826 ENDING AT 34138
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT27.ddl

```
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 34139 ENDING AT 35451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT28.ddl

```
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 35452 ENDING AT 36764
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT29.ddl

```
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 36765 ENDING AT 38077
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT2.ddl

```
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 1314 ENDING AT 2626
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT30.ddl

```
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 38078 ENDING AT 39390
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT31.ddl

```
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 39391 ENDING AT 40703
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT32.ddl

```
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 40704 ENDING AT 42016
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT3.ddl

```
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 2627 ENDING AT 3939
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT4.ddl

```
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 3940 ENDING AT 5252
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT5.ddl

```
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 5253 ENDING AT 6565
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT6.ddl

```
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 6566 ENDING AT 7878
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT7.ddl

```
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 7879 ENDING AT 9191
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT8.ddl

```
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 9192 ENDING AT 10504
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_DISTRICT9.ddl

```
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 10505 ENDING AT 11817
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_HISTORY10.ddl

```
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;
```

CRTB_HISTORY11.ddl

```
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;
```

CRTB_HISTORY12.ddl

```
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;
```

CRTB_HISTORY13.ddl

```
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;
```

CRTB_HISTORY14.ddl

```
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;
```

CRTB_HISTORY15.ddl

```
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;
```

CRTB_HISTORY16.ddl

```
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;
```

CRTB_HISTORY17.ddl

```
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;
```

CRTB_HISTORY18.ddl

```
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;
```

CRTB_HISTORY19.ddl

```
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;
```

CRTB_HISTORY1.ddl

```
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;
```

CRTB_HISTORY20.ddl

```
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;
```

CRTB_HISTORY21.ddl

```
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;
```

CRTB_HISTORY22.ddl

```
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;
```

CRTB_HISTORY23.ddl

```
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;
```

CRTB_HISTORY24.ddl

```
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;
```

CRTB_HISTORY25.ddl

```
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;
```

CRTB_HISTORY26.ddl

```
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;
```

CRTB_HISTORY27.ddl

```
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;
```

CRTB_HISTORY28.ddl

```
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;
```

CRTB_HISTORY29.ddl

```
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;
```

CRTB_HISTORY2.ddl

```
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;
```

CRTB_HISTORY30.ddl

```
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;
```

CRTB_HISTORY31.ddl

```
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;
```

CRTB_HISTORY32.ddl

```
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;
```

CRTB_HISTORY3.ddl

```
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;
```

CRTB_HISTORY4.ddl

```
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;
```

CRTB_HISTORY5.ddl

```
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;
```

CRTB_HISTORY6.ddl

```
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;
```

CRTB_HISTORY7.ddl

```
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;
```

CRTB_HISTORY8.ddl

```
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;
```

CRTB_HISTORY9.ddl

```
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;
```

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;
```

CRTB_NEW_ORDERA10.ddl

```
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 11818 ENDING AT 13130,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA11.ddl

```
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 13131 ENDING AT 14443,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA12.ddl

```
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 14444 ENDING AT 15756,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA13.ddl

```
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 15757 ENDING AT 17069,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA14.ddl

```
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 17070 ENDING AT 18382,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA15.ddl

```
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 18383 ENDING AT 19695,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA16.ddl

```
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 19696 ENDING AT 21008,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA17.ddl

```
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 21009 ENDING AT 22321,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA18.ddl

```
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 22322 ENDING AT 23634,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA19.ddl

```
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 23635 ENDING AT 24947,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA1.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 1313,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA20.ddl

```
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 24948 ENDING AT 26260,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA21.ddl

```
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 26261 ENDING AT 27573,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 1900 ENDING AT 3675
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA22.ddl

```
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 27574 ENDING AT 28886,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA23.ddl

```
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 28887 ENDING AT 30199,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA24.ddl

```
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 30200 ENDING AT 31512,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA25.ddl

```
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 31513 ENDING AT 32825,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA26.ddl

```
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 32826 ENDING AT 34138,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA27.ddl

```
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 34139 ENDING AT 35451,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA28.ddl

```
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 35452 ENDING AT 36764,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA29.ddl

```
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 36765 ENDING AT 38077,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA2.ddl

```
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 1314 ENDING AT 2626,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 1900 ENDING AT 3675
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA30.ddl

```
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 38078 ENDING AT 39390,
```

```
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA31.ddl

```
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 39391 ENDING AT 40703,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA32.ddl

```
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 40704 ENDING AT 42016,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA3.ddl

```
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 2627 ENDING AT 3939,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA4.ddl

```
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 3940 ENDING AT 5252,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA5.ddl

```
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 5253 ENDING AT 6565,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA6.ddl

```
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 6566 ENDING AT 7878,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA7.ddl

```
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 7879 ENDING AT 9191,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA8.ddl

```
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 9192 ENDING AT 10504,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA9.ddl

```
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 10505 ENDING AT 11817,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 1900 ENDING AT 3675
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB10.ddl

```
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 11818 ENDING AT 13130,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB11.ddl

```
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 13131 ENDING AT 14443,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB12.ddl

```
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 14444 ENDING AT 15756,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB13.ddl

```
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 15757 ENDING AT 17069,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB14.ddl

```
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 17070 ENDING AT 18382,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB15.ddl

```
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 18383 ENDING AT 19695,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB16.ddl

```
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 19696 ENDING AT 21008,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB17.ddl

```
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 21009 ENDING AT 22321,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB18.ddl

```
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 22322 ENDING AT 23634,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB19.ddl

```
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 23635 ENDING AT 24947,
                 NO_D_ID STARTING FROM 1 ENDING AT 10,
                 NO_O_ID STARTING FROM 3676 ENDING AT 5451
                )
                ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERB1.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 1313,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB20.ddl

```
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 24948 ENDING AT 26260,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB21.ddl

```
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 26261 ENDING AT 27573,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB22.ddl

```
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 27574 ENDING AT 28886,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB23.ddl

```
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 28887 ENDING AT 30199,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB24.ddl

```
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 30200 ENDING AT 31512,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB25.ddl

```
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 31513 ENDING AT 32825,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB26.ddl

```
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 32826 ENDING AT 34138,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB27.ddl

```
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 34139 ENDING AT 35451,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB28.ddl

```
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 35452 ENDING AT 36764,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB29.ddl

```
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 36765 ENDING AT 38077,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB2.ddl

```
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 1314 ENDING AT 2626,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB30.ddl

```
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 38078 ENDING AT 39390,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB31.ddl

```
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 39391 ENDING AT 40703,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB32.ddl

```
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 40704 ENDING AT 42016,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB3.ddl

```
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 2627 ENDING AT 3939,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB4.ddl

```
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 3940 ENDING AT 5252,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB5.ddl

```
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 5253 ENDING AT 6565,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB6.ddl

```
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 6566 ENDING AT 7878,
                     NO_D_ID STARTING FROM 1 ENDING AT 10,
                     NO_O_ID STARTING FROM 3676 ENDING AT 5451
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB7.ddl

```
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 7879 ENDING AT 9191,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB8.ddl

```
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 9192 ENDING AT 10504,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB9.ddl

```
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 10505 ENDING AT 11817,
                   NO_D_ID STARTING FROM 1 ENDING AT 10,
                   NO_O_ID STARTING FROM 3676 ENDING AT 5451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDER_LINE10.ddl

```
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 11818 ENDING AT 13130,
                   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;
```

CRTB_ORDER_LINE11.ddl

```
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 13131 ENDING AT 14443,
                   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;
```

CRTB_ORDER_LINE12.ddl

```
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 14444 ENDING AT 15756,
                   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;
```

CRTB_ORDER_LINE13.ddl

```
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 15757 ENDING AT 17069,
                   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;
```

CRTB_ORDER_LINE14.ddl

```
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 17070 ENDING AT 18382,
                   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;
```

CRTB_ORDER_LINE15.ddl

```
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 18383 ENDING AT 19695,
                   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;
```

CRTB_ORDER_LINE16.ddl

```
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 19696 ENDING AT 21008,
                   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;
```

CRTB_ORDER_LINE17.ddl

```
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 21009 ENDING AT 22321,
                   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;
```

CRTB_ORDER_LINE18.ddl

```
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 22322 ENDING AT 23634,
                   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;
```

CRTB_ORDER_LINE19.ddl

```
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 23635 ENDING AT 24947,
                   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;
```

CRTB_ORDER_LINE1.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 1313,
                   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;
```

CRTB_ORDER_LINE20.ddl

```
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 24948 ENDING AT 26260,
                   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;
```

CRTB_ORDER_LINE21.ddl

```
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 26261 ENDING AT 27573,
                   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;
```

CRTB_ORDER_LINE22.ddl

```
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 27574 ENDING AT 28886,
                   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;
```

CRTB_ORDER_LINE23.ddl

```
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 28887 ENDING AT 30199,
                   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;
```

CRTB_ORDER_LINE24.ddl

```
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 30200 ENDING AT 31512,
                   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;
```

CRTB_ORDER_LINE25.ddl

```
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 31513 ENDING AT 32825,
                   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;
```

CRTB_ORDER_LINE26.ddl

```
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 32826 ENDING AT 34138,
                   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;
```

CRTB_ORDER_LINE27.ddl

```
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 34139 ENDING AT 35451,
                   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;
```

CRTB_ORDER_LINE28.ddl

```
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 35452 ENDING AT 36764,
                   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;
```

CRTB_ORDER_LINE29.ddl

```
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 36765 ENDING AT 38077,
                   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;
```

CRTB_ORDER_LINE2.ddl

```
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 1314 ENDING AT 2626,
                   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;
```

CRTB_ORDER_LINE30.ddl

```
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 38078 ENDING AT 39390,
                   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;
```

CRTB_ORDER_LINE31.ddl

```
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_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_031
                  INDEX IN ts_orderline_031
                  ORGANIZE BY KEY SEQUENCE (
                   OL_W_ID STARTING FROM 39391 ENDING AT 40703,
                   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;
```

CRTB_ORDER_LINE32.ddl

```
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 40704 ENDING AT 42016,
                   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;
```

CRTB_ORDER_LINE3.ddl

```
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 2627 ENDING AT 3939,
                   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;
```

CRTB_ORDER_LINE4.ddl

```
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 3940 ENDING AT 5252,
                 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;
```

CRTB_ORDER_LINE5.ddl

```
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 5253 ENDING AT 6565,
                 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;
```

CRTB_ORDER_LINE6.ddl

```
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 6566 ENDING AT 7878,
                 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;
```

CRTB_ORDER_LINE7.ddl

```
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 7879 ENDING AT 9191,
                 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;
```

CRTB_ORDER_LINE8.ddl

```
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 9192 ENDING AT 10504,
                 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;
```

CRTB_ORDER_LINE9.ddl

```
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 10505 ENDING AT 11817,
                 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;
```

CRTB_ORDERS10.ddl

```
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 11818 ENDING AT 13130,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS11.ddl

```
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 13131 ENDING AT 14443,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS12.ddl

```
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 14444 ENDING AT 15756,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS13.ddl

```
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 15757 ENDING AT 17069,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS14.ddl

```
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 17070 ENDING AT 18382,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS15.ddl

```
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 18383 ENDING AT 19695,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS16.ddl

```
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 19696 ENDING AT 21008,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS17.ddl

```
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 21009 ENDING AT 22321,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS18.ddl

```
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 22322 ENDING AT 23634,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS19.ddl

```
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 23635 ENDING AT 24947,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS1.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 1313,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS20.ddl

```
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 24948 ENDING AT 26260,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS21.ddl

```
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 26261 ENDING AT 27573,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS22.ddl

```
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 27574 ENDING AT 28886,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS23.ddl

```
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 28887 ENDING AT 30199,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS24.ddl

```
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 30200 ENDING AT 31512,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS25.ddl

```
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 31513 ENDING AT 32825,
                 O_D_ID STARTING FROM 1 ENDING AT 10
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS26.ddl

```
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 32826 ENDING AT 34138,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS27.ddl

```
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 34139 ENDING AT 35451,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS28.ddl

```
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 35452 ENDING AT 36764,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS29.ddl

```
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 36765 ENDING AT 38077,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS2.ddl

```
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 1314 ENDING AT 2626,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS30.ddl

```
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 38078 ENDING AT 39390,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS31.ddl

```
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 39391 ENDING AT 40703,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS32.ddl

```
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 40704 ENDING AT 42016,
                     O_D_ID STARTING FROM 1 ENDING AT 10
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS3.ddl

```
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 2627 ENDING AT 3939,
                   O_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS4.ddl

```
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 3940 ENDING AT 5252,
                   O_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS5.ddl

```
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 5253 ENDING AT 6565,
                   O_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS6.ddl

```
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 6566 ENDING AT 7878,
                   O_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS7.ddl

```
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 7879 ENDING AT 9191,
                   O_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS8.ddl

```
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 9192 ENDING AT 10504,
                   O_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_ORDERS9.ddl

```
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 10505 ENDING AT 11817,
                   O_D_ID STARTING FROM 1 ENDING AT 10
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK10.ddl

```
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 11818 ENDING AT 13130
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK11.ddl

```
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 13131 ENDING AT 14443
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK12.ddl

```
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 14444 ENDING AT 15756
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK13.ddl

```
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 15757 ENDING AT 17069
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK14.ddl

```
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 17070 ENDING AT 18382
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK15.ddl

```
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 18383 ENDING AT 19695
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK16.ddl

```
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 19696 ENDING AT 21008
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK17.ddl

```
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 21009 ENDING AT 22321
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK18.ddl

```
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 22322 ENDING AT 23634
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK19.ddl

```
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 23635 ENDING AT 24947
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK1.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 1313
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK20.ddl

```
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 24948 ENDING AT 26260
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK21.ddl

```
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 26261 ENDING AT 27573
                )
                ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_STOCK22.ddl

```
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 27574 ENDING AT 28886
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK23.ddl

```
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 28887 ENDING AT 30199
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK24.ddl

```
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 30200 ENDING AT 31512
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK25.ddl

```
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 31513 ENDING AT 32825
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK26.ddl

```
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 32826 ENDING AT 34138
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK27.ddl

```
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 34139 ENDING AT 35451
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK28.ddl

```
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 35452 ENDING AT 36764
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK29.ddl

```
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 36765 ENDING AT 38077
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK2.ddl

```
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 1314 ENDING AT 2626
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK30.ddl

```
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 38078 ENDING AT 39390
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK31.ddl

```
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 39391 ENDING AT 40703
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK32.ddl

```
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 40704 ENDING AT 42016
                  )
                  ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK3.ddl

```
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 2627 ENDING AT 3939
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK4.ddl

```
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 3940 ENDING AT 5252
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK5.ddl

```
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 5253 ENDING AT 6565
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK6.ddl

```
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 6566 ENDING AT 7878
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK7.ddl

```
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 7879 ENDING AT 9191
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK8.ddl

```
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 9192 ENDING AT 10504
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK9.ddl

```
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 10505 ENDING AT 11817
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE10.ddl

```
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 11818 ENDING AT 13130
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE11.ddl

```
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 13131 ENDING AT 14443
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE12.ddl

```
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 14444 ENDING AT 15756
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE13.ddl

```
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 15757 ENDING AT 17069
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE14.ddl

```
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 17070 ENDING AT 18382
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE15.ddl

```
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 18383 ENDING AT 19695
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE16.ddl

```
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 19696 ENDING AT 21008
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE17.ddl

```
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 21009 ENDING AT 22321
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE18.ddl

```
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 22322 ENDING AT 23634
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE19.ddl

```
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 23635 ENDING AT 24947
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE1.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 1313
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE20.ddl

```
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 24948 ENDING AT 26260
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE21.ddl

```
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 26261 ENDING AT 27573
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE22.ddl

```
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 27574 ENDING AT 28886
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE23.ddl

```
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 28887 ENDING AT 30199
                    )
                    ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE24.ddl

```
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 30200 ENDING AT 31512
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE25.ddl

```
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 31513 ENDING AT 32825
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE26.ddl

```
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 32826 ENDING AT 34138
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE27.ddl

```
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 34139 ENDING AT 35451
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE28.ddl

```
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 35452 ENDING AT 36764
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE29.ddl

```
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 36765 ENDING AT 38077
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE2.ddl

```
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 1314 ENDING AT 2626
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE30.ddl

```
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 38078 ENDING AT 39390
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE31.ddl

```
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 39391 ENDING AT 40703
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE32.ddl

```
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 40704 ENDING AT 42016
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE3.ddl

```
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 2627 ENDING AT 3939
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE4.ddl

```
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 3940 ENDING AT 5252
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE5.ddl

```
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 5253 ENDING AT 6565
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE6.ddl

```
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 6566 ENDING AT 7878
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE7.ddl

```
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 7879 ENDING AT 9191
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE8.ddl

```
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 9192 ENDING AT 10504
                )
                ALLOW OVERFLOW;
connect reset;
```

CRTB_WAREHOUSE9.ddl

```
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 10505 ENDING AT 11817
                )
                ALLOW OVERFLOW;
connect reset;
```

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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

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_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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

GEN_CUSTOMER_10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11818 13130 -f1
/flx8/flat_010/customer_010_1.dat
```

GEN_CUSTOMER_11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13131 14443 -f1
/flx8/flat_011/customer_011_1.dat
```

GEN_CUSTOMER_12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14444 15756 -f1
/flx8/flat_012/customer_012_1.dat
```

GEN_CUSTOMER_13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15757 17069 -f1
/flx8/flat_013/customer_013_1.dat
```

GEN_CUSTOMER_14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17070 18382 -f1
/flx8/flat_014/customer_014_1.dat
```

GEN_CUSTOMER_15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18383 19695 -f1
/flx8/flat_015/customer_015_1.dat
```

GEN_CUSTOMER_16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19696 21008 -f1
/flx8/flat_016/customer_016_1.dat
```

GEN_CUSTOMER_17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21009 22321 -f1
/flx8/flat_017/customer_017_1.dat
```

GEN_CUSTOMER_18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22322 23634 -f1
/flx8/flat_018/customer_018_1.dat
```

GEN_CUSTOMER_19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23635 24947 -f1
/flx8/flat_019/customer_019_1.dat
```

GEN_CUSTOMER_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 1313 -f1
/flx8/flat_001/customer_001_1.dat
```

GEN_CUSTOMER_20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24948 26260 -f1
/flx8/flat_020/customer_020_1.dat
```

GEN_CUSTOMER_21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26261 27573 -f1
/flx8/flat_021/customer_021_1.dat
```

GEN_CUSTOMER_22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27574 28886 -f1
/flx8/flat_022/customer_022_1.dat
```

GEN_CUSTOMER_23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28887 30199 -f1
/flx8/flat_023/customer_023_1.dat
```

GEN_CUSTOMER_24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30200 31512 -f1
/flx8/flat_024/customer_024_1.dat
```

GEN_CUSTOMER_25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31513 32825 -f1
/flx8/flat_025/customer_025_1.dat
```

GEN_CUSTOMER_26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32826 34138 -f1
/flx8/flat_026/customer_026_1.dat
```

GEN_CUSTOMER_27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34139 35451 -f1
/flx8/flat_027/customer_027_1.dat
```

GEN_CUSTOMER_28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35452 36764 -f1
/flx8/flat_028/customer_028_1.dat
```

GEN_CUSTOMER_29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36765 38077 -f1
/flx8/flat_029/customer_029_1.dat
```

GEN_CUSTOMER_2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1314 2626 -f1
/flx8/flat_002/customer_002_1.dat
```

GEN_CUSTOMER_30.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38078 39390 -f1
/flx8/flat_030/customer_030_1.dat
```

GEN_CUSTOMER_31.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39391 40703 -f1
/flx8/flat_031/customer_031_1.dat
```

GEN_CUSTOMER_32.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40704 42016 -f1
/flx8/flat_032/customer_032_1.dat
```

GEN_CUSTOMER_3.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2627 3939 -f1
/flx8/flat_003/customer_003_1.dat
```

GEN_CUSTOMER_4.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3940 5252 -f1
/flx8/flat_004/customer_004_1.dat
```

GEN_CUSTOMER_5.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5253 6565 -f1
/flx8/flat_005/customer_005_1.dat
```

GEN_CUSTOMER_6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6566 7878 -f1
/flx8/flat_006/customer_006_1.dat
```

GEN_CUSTOMER_7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7879 9191 -f1
/flx8/flat_007/customer_007_1.dat
```

GEN_CUSTOMER_8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9192 10504 -f1
/flx8/flat_008/customer_008_1.dat
```

GEN_CUSTOMER_9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10505 11817 -f1
/flx8/flat_009/customer_009_1.dat
```

GEN_DISTRICT_10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 11818 13130 -f1
/flx8/flat_010/district_010_1.dat
```

GEN_DISTRICT_11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 13131 14443 -f1
/flx8/flat_011/district_011_1.dat
```

GEN_DISTRICT_12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 14444 15756 -f1
/flx8/flat_012/district_012_1.dat
```

GEN_DISTRICT_13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 15757 17069 -f1
/flx8/flat_013/district_013_1.dat
```

GEN_DISTRICT_14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 17070 18382 -f1
/flx8/flat_014/district_014_1.dat
```

GEN_DISTRICT_15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 18383 19695 -f1
/flx8/flat_015/district_015_1.dat
```

GEN_DISTRICT_16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 19696 21008 -f1
/flx8/flat_016/district_016_1.dat
```

GEN_DISTRICT_17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 21009 22321 -f1
/flx8/flat_017/district_017_1.dat
```

GEN_DISTRICT_18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 22322 23634 -f1
/flx8/flat_018/district_018_1.dat
```

GEN_DISTRICT_19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 23635 24947 -f1
/flx8/flat_019/district_019_1.dat
```

GEN_DISTRICT_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1 1313 -f1
/flx8/flat_001/district_001_1.dat
```

GEN_DISTRICT_20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 24948 26260 -f1
/flx8/flat_020/district_020_1.dat
```

GEN_DISTRICT_21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 26261 27573 -f1
/flx8/flat_021/district_021_1.dat
```

GEN_DISTRICT_22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 27574 28886 -f1
/flx8/flat_022/district_022_1.dat
```

GEN_DISTRICT_23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 28887 30199 -f1
/flx8/flat_023/district_023_1.dat
```

GEN_DISTRICT_24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 30200 31512 -f1
/flx8/flat_024/district_024_1.dat
```

GEN_DISTRICT_25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 31513 32825 -f1
/flx8/flat_025/district_025_1.dat
```

GEN_DISTRICT_26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 32826 34138 -f1
/flx8/flat_026/district_026_1.dat
```

GEN_DISTRICT_27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 34139 35451 -f1
/flx8/flat_027/district_027_1.dat
```

GEN_DISTRICT_28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 35452 36764 -f1
/flx8/flat_028/district_028_1.dat
```

GEN_DISTRICT_29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 36765 38077 -f1
/flx8/flat_029/district_029_1.dat
```

GEN_DISTRICT_2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1314 2626 -f1
/flx8/flat_002/district_002_1.dat
```

GEN_DISTRICT_30.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 38078 39390 -f1
/flx8/flat_030/district_030_1.dat
```

GEN_DISTRICT_31.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 39391 40703 -f1
/flx8/flat_031/district_031_1.dat
```

GEN_DISTRICT_32.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 40704 42016 -f1
/flx8/flat_032/district_032_1.dat
```

GEN_DISTRICT_3.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 2627 3939 -f1
/flx8/flat_003/district_003_1.dat
```

GEN_DISTRICT_4.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 3940 5252 -f1
/flx8/flat_004/district_004_1.dat
```

GEN_DISTRICT_5.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 5253 6565 -f1
/flx8/flat_005/district_005_1.dat
```

GEN_DISTRICT_6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 6566 7878 -f1
/flx8/flat_006/district_006_1.dat
```

GEN_DISTRICT_7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 7879 9191 -f1
/flx8/flat_007/district_007_1.dat
```

GEN_DISTRICT_8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 9192 10504 -f1
/flx8/flat_008/district_008_1.dat
```

GEN_DISTRICT_9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 10505 11817 -f1
/flx8/flat_009/district_009_1.dat
```

GEN_HISTORY_10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 11818 13130 -f1
/flx8/flat_010/history_010_1.dat
```

GEN_HISTORY_11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 13131 14443 -f1
/flx8/flat_011/history_011_1.dat
```

GEN_HISTORY_12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 14444 15756 -f1
/flx8/flat_012/history_012_1.dat
```

GEN_HISTORY_13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 15757 17069 -f1
/flx8/flat_013/history_013_1.dat
```

GEN_HISTORY_14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 17070 18382 -f1
/flx8/flat_014/history_014_1.dat
```

GEN_HISTORY_15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 18383 19695 -f1
/flx8/flat_015/history_015_1.dat
```

GEN_HISTORY_16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 19696 21008 -f1
/flx8/flat_016/history_016_1.dat
```

GEN_HISTORY_17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 21009 22321 -f1
/flx8/flat_017/history_017_1.dat
```

GEN_HISTORY_18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 22322 23634 -f1
/flx8/flat_018/history_018_1.dat
```

GEN_HISTORY_19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 23635 24947 -f1
/flx8/flat_019/history_019_1.dat
```

GEN_HISTORY_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1 1313 -f1
/flx8/flat_001/history_001_1.dat
```

GEN_HISTORY_20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 24948 26260 -f1
/flx8/flat_020/history_020_1.dat
```

GEN_HISTORY_21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 26261 27573 -f1
/flx8/flat_021/history_021_1.dat
```

GEN_HISTORY_22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 27574 28886 -f1
/flx8/flat_022/history_022_1.dat
```

GEN_HISTORY_23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 28887 30199 -f1
/flx8/flat_023/history_023_1.dat
```

GEN_HISTORY_24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 30200 31512 -f1
/flx8/flat_024/history_024_1.dat
```

GEN_HISTORY_25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 31513 32825 -f1
/flx8/flat_025/history_025_1.dat
```

GEN_HISTORY_26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 32826 34138 -f1
/flx8/flat_026/history_026_1.dat
```

GEN_HISTORY_27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 34139 35451 -f1
/flx8/flat_027/history_027_1.dat
```

GEN_HISTORY_28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 35452 36764 -f1
/flx8/flat_028/history_028_1.dat
```

GEN_HISTORY_29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 36765 38077 -f1
/flx8/flat_029/history_029_1.dat
```

GEN_HISTORY_2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1314 2626 -f1
/flx8/flat_002/history_002_1.dat
```

GEN_HISTORY_30.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 38078 39390 -f1
/flx8/flat_030/history_030_1.dat
```

GEN_HISTORY_31.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 39391 40703 -f1
/flx8/flat_031/history_031_1.dat
```

GEN_HISTORY_32.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 40704 42016 -f1
/flx8/flat_032/history_032_1.dat
```

GEN_HISTORY_3.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 2627 3939 -f1
/flx8/flat_003/history_003_1.dat
```

GEN_HISTORY_4.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 3940 5252 -f1
/flx8/flat_004/history_004_1.dat
```

GEN_HISTORY_5.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 5253 6565 -f1
/flx8/flat_005/history_005_1.dat
```

GEN_HISTORY_6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 6566 7878 -f1
/flx8/flat_006/history_006_1.dat
```

GEN_HISTORY_7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 7879 9191 -f1
/flx8/flat_007/history_007_1.dat
```

GEN_HISTORY_8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 9192 10504 -f1
/flx8/flat_008/history_008_1.dat
```

GEN_HISTORY_9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 10505 11817 -f1
/flx8/flat_009/history_009_1.dat
```

GEN_ITEM_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 5 -f1 /flx8/flat/item_1.dat
```

GEN_NEW_ORDER_10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 11818 13130 -f1
/flx8/flat_010/neworder_010_1.dat
```

GEN_NEW_ORDER_11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 13131 14443 -f1
/flx8/flat_011/neworder_011_1.dat
```

GEN_NEW_ORDER_12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 14444 15756 -f1
/flx8/flat_012/neworder_012_1.dat
```

GEN_NEW_ORDER_13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 15757 17069 -f1
/flx8/flat_013/neworder_013_1.dat
```

GEN_NEW_ORDER_14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 17070 18382 -f1
/flx8/flat_014/neworder_014_1.dat
```

GEN_NEW_ORDER_15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 18383 19695 -f1
/flx8/flat_015/neworder_015_1.dat
```

GEN_NEW_ORDER_16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 19696 21008 -f1
/flx8/flat_016/neworder_016_1.dat
```

GEN_NEW_ORDER_17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 21009 22321 -f1
/flx8/flat_017/neworder_017_1.dat
```

GEN_NEW_ORDER_18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 22322 23634 -f1
/flx8/flat_018/neworder_018_1.dat
```

GEN_NEW_ORDER_19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 23635 24947 -f1
/flx8/flat_019/neworder_019_1.dat
```

GEN_NEW_ORDER_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1 1313 -f1
/flx8/flat_001/neworder_001_1.dat
```

GEN_NEW_ORDER_20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 24948 26260 -f1
/flx8/flat_020/neworder_020_1.dat
```

GEN_NEW_ORDER_21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 26261 27573 -f1
/flx8/flat_021/neworder_021_1.dat
```

GEN_NEW_ORDER_22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 27574 28886 -f1
/flx8/flat_022/neworder_022_1.dat
```

GEN_NEW_ORDER_23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 28887 30199 -f1
/flx8/flat_023/neworder_023_1.dat
```

GEN_NEW_ORDER_24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 30200 31512 -f1
/flx8/flat_024/neworder_024_1.dat
```

GEN_NEW_ORDER_25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 31513 32825 -f1
/flx8/flat_025/neworder_025_1.dat
```

GEN_NEW_ORDER_26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 32826 34138 -f1
/flx8/flat_026/neworder_026_1.dat
```

GEN_NEW_ORDER_27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 34139 35451 -f1
/flx8/flat_027/neworder_027_1.dat
```

GEN_NEW_ORDER_28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 35452 36764 -f1
/flx8/flat_028/neworder_028_1.dat
```

GEN_NEW_ORDER_29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 36765 38077 -f1
/flx8/flat_029/neworder_029_1.dat
```

GEN_NEW_ORDER_2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1314 2626 -f1
/flx8/flat_002/neworder_002_1.dat
```

GEN_NEW_ORDER_30.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 38078 39390 -f1
/flx8/flat_030/neworder_030_1.dat
```

GEN_NEW_ORDER_31.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 39391 40703 -f1
/flx8/flat_031/neworder_031_1.dat
```

GEN_NEW_ORDER_32.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 40704 42016 -f1
/flx8/flat_032/neworder_032_1.dat
```

GEN_NEW_ORDER_3.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 2627 3939 -f1
/flx8/flat_003/neworder_003_1.dat
```

GEN_NEW_ORDER_4.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 3940 5252 -f1
/flx8/flat_004/neworder_004_1.dat
```

GEN_NEW_ORDER_5.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 5253 6565 -f1
/flx8/flat_005/neworder_005_1.dat
```

GEN_NEW_ORDER_6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 6566 7878 -f1
/flx8/flat_006/neworder_006_1.dat
```

GEN_NEW_ORDER_7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 7879 9191 -f1
/flx8/flat_007/neworder_007_1.dat
```

GEN_NEW_ORDER_8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 9192 10504 -f1
/flx8/flat_008/neworder_008_1.dat
```

GEN_NEW_ORDER_9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 10505 11817 -f1
/flx8/flat_009/neworder_009_1.dat
```

GEN_ORDERS_10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11818 13130 -f1
/flx8/flat_010/orders_010_1.dat -f2
/flx8/flat_010/orderline_010_1.dat
```

GEN_ORDERS_11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13131 14443 -f1
/flx8/flat_011/orders_011_1.dat -f2
/flx8/flat_011/orderline_011_1.dat
```

GEN_ORDERS_12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14444 15756 -f1
/flx8/flat_012/orders_012_1.dat -f2
/flx8/flat_012/orderline_012_1.dat
```

GEN_ORDERS_13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15757 17069 -f1
/flx8/flat_013/orders_013_1.dat -f2
/flx8/flat_013/orderline_013_1.dat
```

GEN_ORDERS_14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17070 18382 -f1
/flx8/flat_014/orders_014_1.dat -f2
/flx8/flat_014/orderline_014_1.dat
```

GEN_ORDERS_15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18383 19695 -f1
/flx8/flat_015/orders_015_1.dat -f2
/flx8/flat_015/orderline_015_1.dat
```

GEN_ORDERS_16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19696 21008 -f1
/flx8/flat_016/orders_016_1.dat -f2
/flx8/flat_016/orderline_016_1.dat
```

GEN_ORDERS_17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21009 22321 -f1
/flx8/flat_017/orders_017_1.dat -f2
/flx8/flat_017/orderline_017_1.dat
```

GEN_ORDERS_18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22322 23634 -f1
/flx8/flat_018/orders_018_1.dat -f2
/flx8/flat_018/orderline_018_1.dat
```

GEN_ORDERS_19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23635 24947 -f1
/flx8/flat_019/orders_019_1.dat -f2
/flx8/flat_019/orderline_019_1.dat
```

GEN_ORDERS_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1 1313 -f1
/flx8/flat_001/orders_001_1.dat -f2
/flx8/flat_001/orderline_001_1.dat
```

GEN_ORDERS_20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24948 26260 -f1
/flx8/flat_020/orders_020_1.dat -f2
/flx8/flat_020/orderline_020_1.dat
```

GEN_ORDERS_21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26261 27573 -f1
/flx8/flat_021/orders_021_1.dat -f2
/flx8/flat_021/orderline_021_1.dat
```

GEN_ORDERS_22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27574 28886 -f1
/flx8/flat_022/orders_022_1.dat -f2
/flx8/flat_022/orderline_022_1.dat
```

GEN_ORDERS_23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28887 30199 -f1
/flx8/flat_023/orders_023_1.dat -f2
/flx8/flat_023/orderline_023_1.dat
```

GEN_ORDERS_24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30200 31512 -f1
/flx8/flat_024/orders_024_1.dat -f2
/flx8/flat_024/orderline_024_1.dat
```

GEN_ORDERS_25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31513 32825 -f1
/flx8/flat_025/orders_025_1.dat -f2
/flx8/flat_025/orderline_025_1.dat
```

GEN_ORDERS_26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32826 34138 -f1
/flx8/flat_026/orders_026_1.dat -f2
/flx8/flat_026/orderline_026_1.dat
```

GEN_ORDERS_27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34139 35451 -f1
/flx8/flat_027/orders_027_1.dat -f2
/flx8/flat_027/orderline_027_1.dat
```

GEN_ORDERS_28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35452 36764 -f1
/flx8/flat_028/orders_028_1.dat -f2
/flx8/flat_028/orderline_028_1.dat
```

GEN_ORDERS_29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36765 38077 -f1
/flx8/flat_029/orders_029_1.dat -f2
/flx8/flat_029/orderline_029_1.dat
```

GEN_ORDERS_2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1314 2626 -f1
/flx8/flat_002/orders_002_1.dat -f2
/flx8/flat_002/orderline_002_1.dat
```

GEN_ORDERS_30.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38078 39390 -f1
/flx8/flat_030/orders_030_1.dat -f2
/flx8/flat_030/orderline_030_1.dat
```

GEN_ORDERS_31.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39391 40703 -f1
/flxnew/flat_031/orders_031_1.dat -f2
/flxnew/flat_031/orderline_031_1.dat
```

GEN_ORDERS_32.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40704 42016 -f1
/flxnew/flat_032/orders_032_1.dat -f2
/flxnew/flat_032/orderline_032_1.dat
```

GEN_ORDERS_3.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2627 3939 -f1
/flx8/flat_003/orders_003_1.dat -f2
/flx8/flat_003/orderline_003_1.dat
```

GEN_ORDERS_4.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3940 5252 -f1
/flx8/flat_004/orders_004_1.dat -f2
/flx8/flat_004/orderline_004_1.dat
```

GEN_ORDERS_5.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5253 6565 -f1
/flx8/flat_005/orders_005_1.dat -f2
/flx8/flat_005/orderline_005_1.dat
```

GEN_ORDERS_6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6566 7878 -f1
/flx8/flat_006/orders_006_1.dat -f2
/flx8/flat_006/orderline_006_1.dat
```

GEN_ORDERS_7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7879 9191 -f1
/flx8/flat_007/orders_007_1.dat -f2
/flx8/flat_007/orderline_007_1.dat
```

GEN_ORDERS_8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9192 10504 -f1
/flx8/flat_008/orders_008_1.dat -f2
/flx8/flat_008/orderline_008_1.dat
```

GEN_ORDERS_9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10505 11817 -f1
/flx8/flat_009/orders_009_1.dat -f2
/flx8/flat_009/orderline_009_1.dat
```

GEN_STOCK_10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11818 13130 -f1
/flx8/flat_010/stock_010_1.dat
```

GEN_STOCK_11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13131 14443 -f1
/flx8/flat_011/stock_011_1.dat
```

GEN_STOCK_12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14444 15756 -f1
/flx8/flat_012/stock_012_1.dat
```

GEN_STOCK_13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15757 17069 -f1
/flx8/flat_013/stock_013_1.dat
```

GEN_STOCK_14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 17070 18382 -f1
/flx8/flat_014/stock_014_1.dat
```

GEN_STOCK_15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18383 19695 -f1
/flx8/flat_015/stock_015_1.dat
```

GEN_STOCK_16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19696 21008 -f1
/flx8/flat_016/stock_016_1.dat
```

GEN_STOCK_17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21009 22321 -f1
/flx8/flat_017/stock_017_1.dat
```

GEN_STOCK_18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22322 23634 -f1
/flx8/flat_018/stock_018_1.dat
```

GEN_STOCK_19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23635 24947 -f1
/flx8/flat_019/stock_019_1.dat
```

GEN_STOCK_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1 1313 -f1
/flx8/flat_001/stock_001_1.dat
```

GEN_STOCK_20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24948 26260 -f1
/flx8/flat_020/stock_020_1.dat
```

GEN_STOCK_21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26261 27573 -f1
/flx8/flat_021/stock_021_1.dat
```

GEN_STOCK_22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27574 28886 -f1
/flx8/flat_022/stock_022_1.dat
```

GEN_STOCK_23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28887 30199 -f1
/flx8/flat_023/stock_023_1.dat
```

GEN_STOCK_24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30200 31512 -f1
/flx8/flat_024/stock_024_1.dat
```

GEN_STOCK_25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31513 32825 -f1
/flx8/flat_025/stock_025_1.dat
```

GEN_STOCK_26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32826 34138 -f1
/flx8/flat_026/stock_026_1.dat
```

GEN_STOCK_27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34139 35451 -f1
/flx8/flat_027/stock_027_1.dat
```

GEN_STOCK_28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 35452 36764 -f1
/flx8/flat_028/stock_028_1.dat
```

GEN_STOCK_29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36765 38077 -f1
/flx8/flat_029/stock_029_1.dat
```

GEN_STOCK_2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1314 2626 -f1
/flx8/flat_002/stock_002_1.dat
```

GEN_STOCK_30.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38078 39390 -f1
/flx8/flat_030/stock_030_1.dat
```

GEN_STOCK_31.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39391 40703 -f1
/flx8/flat_031/stock_031_1.dat
```

GEN_STOCK_32.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40704 42016 -f1
/flx8/flat_032/stock_032_1.dat
```

GEN_STOCK_3.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2627 3939 -f1
/flx8/flat_003/stock_003_1.dat
```

GEN_STOCK_4.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3940 5252 -f1
/flx8/flat_004/stock_004_1.dat
```

GEN_STOCK_5.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5253 6565 -f1
/flx8/flat_005/stock_005_1.dat
```

GEN_STOCK_6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6566 7878 -f1
/flx8/flat_006/stock_006_1.dat
```

GEN_STOCK_7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7879 9191 -f1
/flx8/flat_007/stock_007_1.dat
```

GEN_STOCK_8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9192 10504 -f1
/flx8/flat_008/stock_008_1.dat
```

GEN_STOCK_9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10505 11817 -f1
/flx8/flat_009/stock_009_1.dat
```

GEN_WAREHOUSE_10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 11818 13130 -f1
/flx8/flat_010/warehouse_010_1.dat
```

GEN_WAREHOUSE_11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 13131 14443 -f1
/flx8/flat_011/warehouse_011_1.dat
```

GEN_WAREHOUSE_12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 14444 15756 -f1
/flx8/flat_012/warehouse_012_1.dat
```

GEN_WAREHOUSE_13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 15757 17069 -f1
/flx8/flat_013/warehouse_013_1.dat
```

GEN_WAREHOUSE_14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 17070 18382 -f1
/flx8/flat_014/warehouse_014_1.dat
```

GEN_WAREHOUSE_15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 18383 19695 -f1
/flx8/flat_015/warehouse_015_1.dat
```

GEN_WAREHOUSE_16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 19696 21008 -f1
/flx8/flat_016/warehouse_016_1.dat
```

GEN_WAREHOUSE_17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 21009 22321 -f1
/flx8/flat_017/warehouse_017_1.dat
```

GEN_WAREHOUSE_18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 22322 23634 -f1
/flx8/flat_018/warehouse_018_1.dat
```

GEN_WAREHOUSE_19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 23635 24947 -f1
/flx8/flat_019/warehouse_019_1.dat
```

GEN_WAREHOUSE_1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1 1313 -f1
/flx8/flat_001/warehouse_001_1.dat
```

GEN_WAREHOUSE_20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 24948 26260 -f1
/flx8/flat_020/warehouse_020_1.dat
```

GEN_WAREHOUSE_21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 26261 27573 -f1
/flx8/flat_021/warehouse_021_1.dat
```

GEN_WAREHOUSE_22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 27574 28886 -f1
/flx8/flat_022/warehouse_022_1.dat
```

GEN_WAREHOUSE_23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 28887 30199 -f1
/flx8/flat_023/warehouse_023_1.dat
```

GEN_WAREHOUSE_24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 30200 31512 -f1
/flx8/flat_024/warehouse_024_1.dat
```

GEN_WAREHOUSE_25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 31513 32825 -f1
/flx8/flat_025/warehouse_025_1.dat
```

GEN_WAREHOUSE_26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 32826 34138 -f1
/flx8/flat_026/warehouse_026_1.dat
```

GEN_WAREHOUSE_27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 34139 35451 -f1
/flx8/flat_027/warehouse_027_1.dat
```

GEN_WAREHOUSE_28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 35452 36764 -f1
/flx8/flat_028/warehouse_028_1.dat
```

GEN_WAREHOUSE_29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 36765 38077 -f1
/flx8/flat_029/warehouse_029_1.dat
```

GEN_WAREHOUSE_2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1314 2626 -f1
/flx8/flat_002/warehouse_002_1.dat
```

GEN_WAREHOUSE_30.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 38078 39390 -f1
/flx8/flat_030/warehouse_030_1.dat
```

GEN_WAREHOUSE_31.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 39391 40703 -f1
/flx8/flat_031/warehouse_031_1.dat
```

GEN_WAREHOUSE_32.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 40704 42016 -f1
/flx8/flat_032/warehouse_032_1.dat
```

GEN_WAREHOUSE_3.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 2627 3939 -f1
/flx8/flat_003/warehouse_003_1.dat
```

GEN_WAREHOUSE_4.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 3940 5252 -f1
/flx8/flat_004/warehouse_004_1.dat
```

GEN_WAREHOUSE_5.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 5253 6565 -f1
/flx8/flat_005/warehouse_005_1.dat
```

GEN_WAREHOUSE_6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 6566 7878 -f1
/flx8/flat_006/warehouse_006_1.dat
```

GEN_WAREHOUSE_7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 7879 9191 -f1
/flx8/flat_007/warehouse_007_1.dat
```

GEN_WAREHOUSE_8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 9192 10504 -f1
/flx8/flat_008/warehouse_008_1.dat
```

GEN_WAREHOUSE_9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 10505 11817 -f1
/flx8/flat_009/warehouse_009_1.dat
```

LOAD_CUSTOMER10_1.ddl

```
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 39390000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER11_1.ddl

```
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 39390000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER1_1.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 39390000 INSERT INTO CUSTOMER1;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER12_1.ddl

```
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 39390000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER13_1.ddl

```
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 39390000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER14_1.ddl

```
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 39390000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER15_1.ddl

```
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 39390000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER16_1.ddl

```
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 39390000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER17_1.ddl

```
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 39390000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER18_1.ddl

```
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 39390000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER19_1.ddl

```
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 39390000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER20_1.ddl

```
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 39390000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER21_1.ddl

```
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 39390000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER2_1.ddl

```
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 39390000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER22_1.ddl

```
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 39390000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER23_1.ddl

```
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 39390000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER24_1.ddl

```
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 39390000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER25_1.ddl

```
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 39390000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER26_1.ddl

```
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 39390000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER27_1.ddl

```
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 39390000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER28_1.ddl

```
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 39390000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER29_1.ddl

```
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 39390000 INSERT INTO CUSTOMER29;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER30_1.ddl

```
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 39390000 INSERT INTO CUSTOMER30;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER31_1.ddl

```
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 39390000 INSERT INTO CUSTOMER31;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER3_1.ddl

```
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 39390000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER32_1.ddl

```
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 39390000 INSERT INTO CUSTOMER32;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER4_1.ddl

```
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 39390000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER5_1.ddl

```
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 39390000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER6_1.ddl

```
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 39390000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER7_1.ddl

```
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 39390000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER8_1.ddl

```
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 39390000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
```

LOAD_CUSTOMER9_1.ddl

```
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 39390000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
```

LOAD_DISTRICT10_1.ddl

```
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;
```

LOAD_DISTRICT11_1.ddl

```
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;
```

LOAD_DISTRICT1_1.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;
```

LOAD_DISTRICT12_1.ddl

```
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;
```

LOAD_DISTRICT13_1.ddl

```
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;
```

LOAD_DISTRICT14_1.ddl

```
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;
```

LOAD_DISTRICT15_1.ddl

```
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;
```

LOAD_DISTRICT16_1.ddl

```
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;
```

LOAD_DISTRICT17_1.ddl

```
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;
```

LOAD_DISTRICT18_1.ddl

```
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;
```

LOAD_DISTRICT19_1.ddl

```
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;
```

LOAD_DISTRICT20_1.ddl

```
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;
```

LOAD_DISTRICT21_1.ddl

```
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;
```

LOAD_DISTRICT2_1.ddl

```
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;
```

LOAD_DISTRICT22_1.ddl

```
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;
```

LOAD_DISTRICT23_1.ddl

```
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;
```

LOAD_DISTRICT24_1.ddl

```
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;
```

LOAD_DISTRICT25_1.ddl

```
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;
```

LOAD_DISTRICT26_1.ddl

```
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;
```

LOAD_DISTRICT27_1.ddl

```
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;
```

LOAD_DISTRICT28_1.ddl

```
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;
```

LOAD_DISTRICT29_1.ddl

```
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;
```

LOAD_DISTRICT30_1.ddl

```
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;
```

LOAD_DISTRICT31_1.ddl

```
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;
```

LOAD_DISTRICT3_1.ddl

```
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;
```

LOAD_DISTRICT32_1.ddl

```
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;
```

LOAD_DISTRICT4_1.ddl

```
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;
```

LOAD_DISTRICT5_1.ddl

```
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;
```

LOAD_DISTRICT6_1.ddl

```
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;
```

LOAD_DISTRICT7_1.ddl

```
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;
```

LOAD_DISTRICT8_1.ddl

```
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;
```

LOAD_DISTRICT9_1.ddl

```
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;
```

LOAD_HISTORY10.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY11.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY12.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY13.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY14.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY15.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY16.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY17.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY18.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY19.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY1.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 16 ;
connect reset;
```

LOAD_HISTORY20.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY21.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY22.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY23.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY24.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY25.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY26.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY27.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY28.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY29.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY2.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY30.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY31.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY32.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY3.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY4.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY5.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY6.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY7.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY8.ddl

```
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 16 ;
connect reset;
```

LOAD_HISTORY9.ddl

```
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 16 ;
connect reset;
```

LOAD_ITEM_1.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;
```

LOAD_NEW_ORDERA10_1.ddl

```
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;
```

LOAD_NEW_ORDERA11_1.ddl

```
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;
```

LOAD_NEW_ORDERA1_1.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;
```

LOAD_NEW_ORDERA12_1.ddl

```
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;
```

LOAD_NEW_ORDERA13_1.ddl

```
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;
```

LOAD_NEW_ORDERA14_1.ddl

```
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;
```

LOAD_NEW_ORDERA15_1.ddl

```
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;
```

LOAD_NEW_ORDERA16_1.ddl

```
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;
```

LOAD_NEW_ORDERA17_1.ddl

```
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;
```

LOAD_NEW_ORDERA18_1.ddl

```
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;
```

LOAD_NEW_ORDERA19_1.ddl

```
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;
```

LOAD_NEW_ORDERA20_1.ddl

```
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;
```

LOAD_NEW_ORDERA21_1.ddl

```
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;
```

LOAD_NEW_ORDERA2_1.ddl

```
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;
```

LOAD_NEW_ORDERA22_1.ddl

```
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;
```

LOAD_NEW_ORDERA23_1.ddl

```
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;
```

LOAD_NEW_ORDERA24_1.ddl

```
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;
```

LOAD_NEW_ORDERA25_1.ddl

```
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;
```

LOAD_NEW_ORDERA26_1.ddl

```
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;
```

LOAD_NEW_ORDERA27_1.ddl

```
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;
```

LOAD_NEW_ORDERA28_1.ddl

```
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;
```

LOAD_NEW_ORDERA29_1.ddl

```
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;
```

LOAD_NEW_ORDERA30_1.ddl

```
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;
```

LOAD_NEW_ORDERA31_1.ddl

```
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;
```

LOAD_NEW_ORDERA3_1.ddl

```
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;
```

LOAD_NEW_ORDERA32_1.ddl

```
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;
```

LOAD_NEW_ORDERA4_1.ddl

```
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;
```

LOAD_NEW_ORDERA5_1.ddl

```
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;
```

LOAD_NEW_ORDERA6_1.ddl

```
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;
```

LOAD_NEW_ORDERA7_1.ddl

```
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;
```

LOAD_NEW_ORDERA8_1.ddl

```
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;
```

LOAD_NEW_ORDERA9_1.ddl

```
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;
```

LOAD_ORDER_LINE10_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_010/orderline_010_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE11_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_011/orderline_011_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE1_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_001/orderline_001_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE12_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_012/orderline_012_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE13_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_013/orderline_013_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE14_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_014/orderline_014_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE15_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_015/orderline_015_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE16_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_016/orderline_016_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE17_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_017/orderline_017_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE18_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_018/orderline_018_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE19_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_019/orderline_019_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE20_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_020/orderline_020_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE21_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_021/orderline_021_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE2_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_002/orderline_002_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE22_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_022/orderline_022_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE23_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_023/orderline_023_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE24_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
```

```
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_024/orderline_024_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE25_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_025/orderline_025_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE26_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_026/orderline_026_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE27_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_027/orderline_027_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE28_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_028/orderline_028_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE29_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_029/orderline_029_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE29;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE30_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_030/orderline_030_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE30;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE31_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_031/orderline_031_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE3_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_003/orderline_003_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE32_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_032/orderline_032_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE4_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_004/orderline_004_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE5_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_005/orderline_005_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE6_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_006/orderline_006_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE7_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_007/orderline_007_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE8_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_008/orderline_008_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDER_LINE9_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flxnew/flat_009/orderline_009_1.dat OF DEL MODIFIED
BY   COLDEL|   TIMESTAMPFORMAT="YYYY-MM-DD   HH:MM:SS"   KEEPBLANKS
COMPOUND=50  COMMITCOUNT 433290000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
```

LOAD_ORDERS10_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS11_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS1_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS12_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS13_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS14_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS15_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS16_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS17_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS18_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS19_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS20_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS21_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS2_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS22_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS23_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS24_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS25_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS26_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS27_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS28_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS29_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS30_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS31_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS3_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS32_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS4_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS5_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS6_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS7_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS8_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_ORDERS9_1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flxnew/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;
```

LOAD_STOCK10_1.ddl

```
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 131300000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK11_1.ddl

```
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 131300000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK1_1.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 131300000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK12_1.ddl

```
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 131300000 INSERT INTO STOCK12;
COMMIT WORK;
```

```
CONNECT RESET;
```

LOAD_STOCK13_1.ddl

```
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 131300000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK14_1.ddl

```
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 131300000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK15_1.ddl

```
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 131300000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK16_1.ddl

```
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 131300000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK17_1.ddl

```
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 131300000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK18_1.ddl

```
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 131300000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK19_1.ddl

```
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 131300000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK20_1.ddl

```
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 131300000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK21_1.ddl

```
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 131300000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK2_1.ddl

```
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 131300000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK22_1.ddl

```
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 131300000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK23_1.ddl

```
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 131300000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK24_1.ddl

```
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 131300000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK25_1.ddl

```
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 131300000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK26_1.ddl

```
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 131300000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK27_1.ddl

```
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 131300000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK28_1.ddl

```
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 131300000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK29_1.ddl

```
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 131300000 INSERT INTO STOCK29;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK30_1.ddl

```
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 131300000 INSERT INTO STOCK30;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK31_1.ddl

```
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 131300000 INSERT INTO STOCK31;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK3_1.ddl

```
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 131300000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK32_1.ddl

```
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 131300000 INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK4_1.ddl

```
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 131300000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK5_1.ddl

```
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 131300000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK6_1.ddl

```
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 131300000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK7_1.ddl

```
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 131300000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK8_1.ddl

```
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 131300000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK9_1.ddl

```
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 131300000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
```

LOAD_WAREHOUSE10_1.ddl

```
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;
```

LOAD_WAREHOUSE11_1.ddl

```
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;
```

LOAD_WAREHOUSE1_1.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;
```

LOAD_WAREHOUSE12_1.ddl

```
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;
```

LOAD_WAREHOUSE13_1.ddl

```
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;
```

LOAD_WAREHOUSE14_1.ddl

```
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;
```

LOAD_WAREHOUSE15_1.ddl

```
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;
```

LOAD_WAREHOUSE16_1.ddl

```
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;
```

LOAD_WAREHOUSE17_1.ddl

```
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;
```

LOAD_WAREHOUSE18_1.ddl

```
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;
```

LOAD_WAREHOUSE19_1.ddl

```
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;
```

LOAD_WAREHOUSE20_1.ddl

```
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;
```

LOAD_WAREHOUSE21_1.ddl

```
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;
```

LOAD_WAREHOUSE2_1.ddl

```
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;
```

LOAD_WAREHOUSE22_1.ddl

```
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;
```

LOAD_WAREHOUSE23_1.ddl

```
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;
```

LOAD_WAREHOUSE24_1.ddl

```
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;
```

LOAD_WAREHOUSE25_1.ddl

```
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;
```

LOAD_WAREHOUSE26_1.ddl

```
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;
```

LOAD_WAREHOUSE27_1.ddl

```
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;
```

LOAD_WAREHOUSE28_1.ddl

```
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;
```

LOAD_WAREHOUSE29_1.ddl

```
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;
```

LOAD_WAREHOUSE30_1.ddl

```
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;
```

LOAD_WAREHOUSE31_1.ddl

```
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;
```

LOAD_WAREHOUSE3_1.ddl

```
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;
```

LOAD_WAREHOUSE32_1.ddl

```
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;
```

LOAD_WAREHOUSE4_1.ddl

```
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;
```

LOAD_WAREHOUSE5_1.ddl

```
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;
```

LOAD_WAREHOUSE6_1.ddl

```
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;
```

LOAD_WAREHOUSE7_1.ddl

```
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;
```

LOAD_WAREHOUSE8_1.ddl

```
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;
```

LOAD_WAREHOUSE9_1.ddl

```
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;
```

RNST_CUSTOMER10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_CUSTOMER9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_DISTRICT9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_HISTORY9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ITEM.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ITEM AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERA9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_NEW_ORDERB9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDER_LINE9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_ORDERS9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_STOCK9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE10.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE11.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE12.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE13.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE14.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE15.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE16.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE17.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE18.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE19.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE1.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE20.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE21.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE22.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE23.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE24.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE25.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE26.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE27.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE28.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE29.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE2.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE30.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE31.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE32.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE3.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE4.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE5.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE6.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE7.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE8.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

RNST_WAREHOUSE9.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

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;
```

ts/alter_tablespace.ddl

```
connect to tpcc;

-- Create new bufferpools
create bufferpool ITM       size 100    pagesize 8192  not extended storage;

create bufferpool WDS1      size 100    pagesize 4096;
create bufferpool WDS2      size 100    pagesize 4096;
create bufferpool WDS3      size 100    pagesize 4096;
create bufferpool WDS4      size 100    pagesize 4096;
create bufferpool WDS5      size 100    pagesize 4096;
create bufferpool WDS6      size 100    pagesize 4096;
create bufferpool WDS7      size 100    pagesize 4096;
create bufferpool WDS8      size 100    pagesize 4096;

create bufferpool STK1      size 100    pagesize 4096;
create bufferpool STK2      size 100    pagesize 4096;
create bufferpool STK3      size 100    pagesize 4096;
create bufferpool STK4      size 100    pagesize 4096;
create bufferpool STK5      size 100    pagesize 4096;
create bufferpool STK6      size 100    pagesize 4096;
create bufferpool STK7      size 100    pagesize 4096;
create bufferpool STK8      size 100    pagesize 4096;

create bufferpool CST1      size 100    pagesize 4096;
create bufferpool CST2      size 100    pagesize 4096;
create bufferpool CST3      size 100    pagesize 4096;
create bufferpool CST4      size 100    pagesize 4096;
create bufferpool CST5      size 100    pagesize 4096;
create bufferpool CST6      size 100    pagesize 4096;
create bufferpool CST7      size 100    pagesize 4096;
create bufferpool CST8      size 100    pagesize 4096;

create bufferpool NEW1      size 100    pagesize 4096;
create bufferpool NEW2      size 100    pagesize 4096;
create bufferpool NEW3      size 100    pagesize 4096;
create bufferpool NEW4      size 100    pagesize 4096;
create bufferpool NEW5      size 100    pagesize 4096;
create bufferpool NEW6      size 100    pagesize 4096;
create bufferpool NEW7      size 100    pagesize 4096;
create bufferpool NEW8      size 100    pagesize 4096;

create bufferpool OLNORDIORD1 size 100   pagesize 8192;
create bufferpool OLNORDIORD2 size 100   pagesize 8192;
create bufferpool OLNORDIORD3 size 100   pagesize 8192;
create bufferpool OLNORDIORD4 size 100   pagesize 8192;
create bufferpool OLNORDIORD5 size 100   pagesize 8192;
create bufferpool OLNORDIORD6 size 100   pagesize 8192;
create bufferpool OLNORDIORD7 size 100   pagesize 8192;
create bufferpool OLNORDIORD8 size 100   pagesize 8192;

create bufferpool HST1      size 100    pagesize 16384;
create bufferpool HST2      size 100    pagesize 16384;
create bufferpool HST3      size 100    pagesize 16384;
create bufferpool HST4      size 100    pagesize 16384;
create bufferpool HST5      size 100    pagesize 16384;
create bufferpool HST6      size 100    pagesize 16384;
create bufferpool HST7      size 100    pagesize 16384;
create bufferpool HST8      size 100    pagesize 16384;

create bufferpool CSTI1     size 100    pagesize 8192;
create bufferpool CSTI2     size 100    pagesize 8192;
create bufferpool CSTI3     size 100    pagesize 8192;
create bufferpool CSTI4     size 100    pagesize 8192;
create bufferpool CSTI5     size 100    pagesize 8192;
create bufferpool CSTI6     size 100    pagesize 8192;
create bufferpool CSTI7     size 100    pagesize 8192;
create bufferpool CSTI8     size 100    pagesize 8192;

-- Alter tablespaces to use these bufferpools
```

```
alter tablespace ts_wh_001 bufferpool WDS1;
alter tablespace ts_wh_002 bufferpool WDS1;
alter tablespace ts_wh_003 bufferpool WDS1;
alter tablespace ts_wh_004 bufferpool WDS1;
alter tablespace ts_wh_005 bufferpool WDS2;
alter tablespace ts_wh_006 bufferpool WDS2;
alter tablespace ts_wh_007 bufferpool WDS2;
alter tablespace ts_wh_008 bufferpool WDS2;
alter tablespace ts_wh_009 bufferpool WDS3;
alter tablespace ts_wh_010 bufferpool WDS3;
alter tablespace ts_wh_011 bufferpool WDS3;
alter tablespace ts_wh_012 bufferpool WDS3;
alter tablespace ts_wh_013 bufferpool WDS4;
alter tablespace ts_wh_014 bufferpool WDS4;
alter tablespace ts_wh_015 bufferpool WDS4;
alter tablespace ts_wh_016 bufferpool WDS4;
alter tablespace ts_wh_017 bufferpool WDS5;
alter tablespace ts_wh_018 bufferpool WDS5;
alter tablespace ts_wh_019 bufferpool WDS5;
alter tablespace ts_wh_020 bufferpool WDS5;
alter tablespace ts_wh_021 bufferpool WDS6;
alter tablespace ts_wh_022 bufferpool WDS6;
alter tablespace ts_wh_023 bufferpool WDS6;
alter tablespace ts_wh_024 bufferpool WDS6;
alter tablespace ts_wh_025 bufferpool WDS7;
alter tablespace ts_wh_026 bufferpool WDS7;
alter tablespace ts_wh_027 bufferpool WDS7;
alter tablespace ts_wh_028 bufferpool WDS7;
alter tablespace ts_wh_029 bufferpool WDS8;
alter tablespace ts_wh_030 bufferpool WDS8;
alter tablespace ts_wh_031 bufferpool WDS8;
alter tablespace ts_wh_032 bufferpool WDS8;


alter tablespace ts_dis_001 bufferpool WDS1;
alter tablespace ts_dis_002 bufferpool WDS1;
alter tablespace ts_dis_003 bufferpool WDS1;
alter tablespace ts_dis_004 bufferpool WDS1;
alter tablespace ts_dis_005 bufferpool WDS2;
alter tablespace ts_dis_006 bufferpool WDS2;
alter tablespace ts_dis_007 bufferpool WDS2;
alter tablespace ts_dis_008 bufferpool WDS2;
alter tablespace ts_dis_009 bufferpool WDS3;
alter tablespace ts_dis_010 bufferpool WDS3;
alter tablespace ts_dis_011 bufferpool WDS3;
alter tablespace ts_dis_012 bufferpool WDS3;
alter tablespace ts_dis_013 bufferpool WDS4;
alter tablespace ts_dis_014 bufferpool WDS4;
alter tablespace ts_dis_015 bufferpool WDS4;
alter tablespace ts_dis_016 bufferpool WDS4;
alter tablespace ts_dis_017 bufferpool WDS5;
alter tablespace ts_dis_018 bufferpool WDS5;
alter tablespace ts_dis_019 bufferpool WDS5;
alter tablespace ts_dis_020 bufferpool WDS5;
alter tablespace ts_dis_021 bufferpool WDS6;
alter tablespace ts_dis_022 bufferpool WDS6;
alter tablespace ts_dis_023 bufferpool WDS6;
alter tablespace ts_dis_024 bufferpool WDS6;
alter tablespace ts_dis_025 bufferpool WDS7;
alter tablespace ts_dis_026 bufferpool WDS7;
alter tablespace ts_dis_027 bufferpool WDS7;
alter tablespace ts_dis_028 bufferpool WDS7;
alter tablespace ts_dis_029 bufferpool WDS8;
alter tablespace ts_dis_030 bufferpool WDS8;
alter tablespace ts_dis_031 bufferpool WDS8;
alter tablespace ts_dis_032 bufferpool WDS8;

alter tablespace ts_item bufferpool ITM;

alter tablespace ts_stock_001 bufferpool STK1;
alter tablespace ts_stock_002 bufferpool STK1;
alter tablespace ts_stock_003 bufferpool STK1;
alter tablespace ts_stock_004 bufferpool STK1;
alter tablespace ts_stock_005 bufferpool STK2;
alter tablespace ts_stock_006 bufferpool STK2;
alter tablespace ts_stock_007 bufferpool STK2;
alter tablespace ts_stock_008 bufferpool STK2;
alter tablespace ts_stock_009 bufferpool STK3;
alter tablespace ts_stock_010 bufferpool STK3;
alter tablespace ts_stock_011 bufferpool STK3;
alter tablespace ts_stock_012 bufferpool STK3;
alter tablespace ts_stock_013 bufferpool STK4;
alter tablespace ts_stock_014 bufferpool STK4;
alter tablespace ts_stock_015 bufferpool STK4;
alter tablespace ts_stock_016 bufferpool STK4;
alter tablespace ts_stock_017 bufferpool STK5;
alter tablespace ts_stock_018 bufferpool STK5;
alter tablespace ts_stock_019 bufferpool STK5;
alter tablespace ts_stock_020 bufferpool STK5;
alter tablespace ts_stock_021 bufferpool STK6;
alter tablespace ts_stock_022 bufferpool STK6;
alter tablespace ts_stock_023 bufferpool STK6;
alter tablespace ts_stock_024 bufferpool STK6;
alter tablespace ts_stock_025 bufferpool STK7;
alter tablespace ts_stock_026 bufferpool STK7;
alter tablespace ts_stock_027 bufferpool STK7;
alter tablespace ts_stock_028 bufferpool STK7;
alter tablespace ts_stock_029 bufferpool STK8;
alter tablespace ts_stock_030 bufferpool STK8;
alter tablespace ts_stock_031 bufferpool STK8;
alter tablespace ts_stock_032 bufferpool STK8;


alter tablespace ts_customer_001 bufferpool CST1;
alter tablespace ts_customer_002 bufferpool CST1;
alter tablespace ts_customer_003 bufferpool CST1;
alter tablespace ts_customer_004 bufferpool CST1;
alter tablespace ts_customer_005 bufferpool CST2;
alter tablespace ts_customer_006 bufferpool CST2;
alter tablespace ts_customer_007 bufferpool CST2;
alter tablespace ts_customer_008 bufferpool CST2;
alter tablespace ts_customer_009 bufferpool CST3;
alter tablespace ts_customer_010 bufferpool CST3;
alter tablespace ts_customer_011 bufferpool CST3;
alter tablespace ts_customer_012 bufferpool CST3;
alter tablespace ts_customer_013 bufferpool CST4;
alter tablespace ts_customer_014 bufferpool CST4;
alter tablespace ts_customer_015 bufferpool CST4;
alter tablespace ts_customer_016 bufferpool CST4;
alter tablespace ts_customer_017 bufferpool CST5;
alter tablespace ts_customer_018 bufferpool CST5;
alter tablespace ts_customer_019 bufferpool CST5;
alter tablespace ts_customer_020 bufferpool CST5;
alter tablespace ts_customer_021 bufferpool CST6;
alter tablespace ts_customer_022 bufferpool CST6;
alter tablespace ts_customer_023 bufferpool CST6;
alter tablespace ts_customer_024 bufferpool CST6;
alter tablespace ts_customer_025 bufferpool CST7;
alter tablespace ts_customer_026 bufferpool CST7;
alter tablespace ts_customer_027 bufferpool CST7;
alter tablespace ts_customer_028 bufferpool CST7;
alter tablespace ts_customer_029 bufferpool CST8;
alter tablespace ts_customer_030 bufferpool CST8;
alter tablespace ts_customer_031 bufferpool CST8;
alter tablespace ts_customer_032 bufferpool CST8;

alter tablespace is_customer_001 bufferpool CSTI1;
alter tablespace is_customer_002 bufferpool CSTI1;
alter tablespace is_customer_003 bufferpool CSTI1;
alter tablespace is_customer_004 bufferpool CSTI1;
alter tablespace is_customer_005 bufferpool CSTI2;
alter tablespace is_customer_006 bufferpool CSTI2;
alter tablespace is_customer_007 bufferpool CSTI2;
alter tablespace is_customer_008 bufferpool CSTI2;
alter tablespace is_customer_009 bufferpool CSTI3;
alter tablespace is_customer_010 bufferpool CSTI3;
alter tablespace is_customer_011 bufferpool CSTI3;
alter tablespace is_customer_012 bufferpool CSTI3;
alter tablespace is_customer_013 bufferpool CSTI4;
alter tablespace is_customer_014 bufferpool CSTI4;
alter tablespace is_customer_015 bufferpool CSTI4;
alter tablespace is_customer_016 bufferpool CSTI4;
alter tablespace is_customer_017 bufferpool CSTI5;
alter tablespace is_customer_018 bufferpool CSTI5;
alter tablespace is_customer_019 bufferpool CSTI5;
alter tablespace is_customer_020 bufferpool CSTI5;
alter tablespace is_customer_021 bufferpool CSTI6;
alter tablespace is_customer_022 bufferpool CSTI6;
alter tablespace is_customer_023 bufferpool CSTI6;
alter tablespace is_customer_024 bufferpool CSTI6;
alter tablespace is_customer_025 bufferpool CSTI7;
alter tablespace is_customer_026 bufferpool CSTI7;
alter tablespace is_customer_027 bufferpool CSTI7;
alter tablespace is_customer_028 bufferpool CSTI7;
alter tablespace is_customer_029 bufferpool CSTI8;
alter tablespace is_customer_030 bufferpool CSTI8;
alter tablespace is_customer_031 bufferpool CSTI8;
alter tablespace is_customer_032 bufferpool CSTI8;

alter tablespace ts_order_001 bufferpool OLNORDIORD1;
alter tablespace ts_order_002 bufferpool OLNORDIORD1;
alter tablespace ts_order_003 bufferpool OLNORDIORD1;
alter tablespace ts_order_004 bufferpool OLNORDIORD1;
alter tablespace ts_order_005 bufferpool OLNORDIORD2;
alter tablespace ts_order_006 bufferpool OLNORDIORD2;
alter tablespace ts_order_007 bufferpool OLNORDIORD2;
alter tablespace ts_order_008 bufferpool OLNORDIORD2;
alter tablespace ts_order_009 bufferpool OLNORDIORD3;
alter tablespace ts_order_010 bufferpool OLNORDIORD3;
alter tablespace ts_order_011 bufferpool OLNORDIORD3;
alter tablespace ts_order_012 bufferpool OLNORDIORD3;
alter tablespace ts_order_013 bufferpool OLNORDIORD4;
alter tablespace ts_order_014 bufferpool OLNORDIORD4;
alter tablespace ts_order_015 bufferpool OLNORDIORD4;
alter tablespace ts_order_016 bufferpool OLNORDIORD4;
alter tablespace ts_order_017 bufferpool OLNORDIORD5;
alter tablespace ts_order_018 bufferpool OLNORDIORD5;
alter tablespace ts_order_019 bufferpool OLNORDIORD5;
alter tablespace ts_order_020 bufferpool OLNORDIORD5;
alter tablespace ts_order_021 bufferpool OLNORDIORD6;
alter tablespace ts_order_022 bufferpool OLNORDIORD6;
alter tablespace ts_order_023 bufferpool OLNORDIORD6;
alter tablespace ts_order_024 bufferpool OLNORDIORD6;
alter tablespace ts_order_025 bufferpool OLNORDIORD7;
alter tablespace ts_order_026 bufferpool OLNORDIORD7;
alter tablespace ts_order_027 bufferpool OLNORDIORD7;
alter tablespace ts_order_028 bufferpool OLNORDIORD7;
alter tablespace ts_order_029 bufferpool OLNORDIORD8;
alter tablespace ts_order_030 bufferpool OLNORDIORD8;
alter tablespace ts_order_031 bufferpool OLNORDIORD8;
alter tablespace ts_order_032 bufferpool OLNORDIORD8;

alter tablespace ts_orderline_001 bufferpool OLNORDIORD1;
alter tablespace ts_orderline_002 bufferpool OLNORDIORD1;
alter tablespace ts_orderline_003 bufferpool OLNORDIORD1;
alter tablespace ts_orderline_004 bufferpool OLNORDIORD1;
alter tablespace ts_orderline_005 bufferpool OLNORDIORD2;
alter tablespace ts_orderline_006 bufferpool OLNORDIORD2;
alter tablespace ts_orderline_007 bufferpool OLNORDIORD2;
alter tablespace ts_orderline_008 bufferpool OLNORDIORD2;
```

```
alter tablespace ts_orderline_009 bufferpool OLNORDIORD3;
alter tablespace ts_orderline_010 bufferpool OLNORDIORD3;
alter tablespace ts_orderline_011 bufferpool OLNORDIORD3;
alter tablespace ts_orderline_012 bufferpool OLNORDIORD3;
alter tablespace ts_orderline_013 bufferpool OLNORDIORD4;
alter tablespace ts_orderline_014 bufferpool OLNORDIORD4;
alter tablespace ts_orderline_015 bufferpool OLNORDIORD4;
alter tablespace ts_orderline_016 bufferpool OLNORDIORD4;
alter tablespace ts_orderline_017 bufferpool OLNORDIORD5;
alter tablespace ts_orderline_018 bufferpool OLNORDIORD5;
alter tablespace ts_orderline_019 bufferpool OLNORDIORD5;
alter tablespace ts_orderline_020 bufferpool OLNORDIORD5;
alter tablespace ts_orderline_021 bufferpool OLNORDIORD6;
alter tablespace ts_orderline_022 bufferpool OLNORDIORD6;
alter tablespace ts_orderline_023 bufferpool OLNORDIORD6;
alter tablespace ts_orderline_024 bufferpool OLNORDIORD6;
alter tablespace ts_orderline_025 bufferpool OLNORDIORD7;
alter tablespace ts_orderline_026 bufferpool OLNORDIORD7;
alter tablespace ts_orderline_027 bufferpool OLNORDIORD7;
alter tablespace ts_orderline_028 bufferpool OLNORDIORD7;
alter tablespace ts_orderline_029 bufferpool OLNORDIORD8;
alter tablespace ts_orderline_030 bufferpool OLNORDIORD8;
alter tablespace ts_orderline_031 bufferpool OLNORDIORD8;
alter tablespace ts_orderline_032 bufferpool OLNORDIORD8;

alter tablespace is_order_001 bufferpool OLNORDIORD1;
alter tablespace is_order_002 bufferpool OLNORDIORD1;
alter tablespace is_order_003 bufferpool OLNORDIORD1;
alter tablespace is_order_004 bufferpool OLNORDIORD1;
alter tablespace is_order_005 bufferpool OLNORDIORD2;
alter tablespace is_order_006 bufferpool OLNORDIORD2;
alter tablespace is_order_007 bufferpool OLNORDIORD2;
alter tablespace is_order_008 bufferpool OLNORDIORD2;
alter tablespace is_order_009 bufferpool OLNORDIORD3;
alter tablespace is_order_010 bufferpool OLNORDIORD3;
alter tablespace is_order_011 bufferpool OLNORDIORD3;
alter tablespace is_order_012 bufferpool OLNORDIORD3;
alter tablespace is_order_013 bufferpool OLNORDIORD4;
alter tablespace is_order_014 bufferpool OLNORDIORD4;
alter tablespace is_order_015 bufferpool OLNORDIORD4;
alter tablespace is_order_016 bufferpool OLNORDIORD4;
alter tablespace is_order_017 bufferpool OLNORDIORD5;
alter tablespace is_order_018 bufferpool OLNORDIORD5;
alter tablespace is_order_019 bufferpool OLNORDIORD5;
alter tablespace is_order_020 bufferpool OLNORDIORD5;
alter tablespace is_order_021 bufferpool OLNORDIORD6;
alter tablespace is_order_022 bufferpool OLNORDIORD6;
alter tablespace is_order_023 bufferpool OLNORDIORD6;
alter tablespace is_order_024 bufferpool OLNORDIORD6;
alter tablespace is_order_025 bufferpool OLNORDIORD7;
alter tablespace is_order_026 bufferpool OLNORDIORD7;
alter tablespace is_order_027 bufferpool OLNORDIORD7;
alter tablespace is_order_028 bufferpool OLNORDIORD7;
alter tablespace is_order_029 bufferpool OLNORDIORD8;
alter tablespace is_order_030 bufferpool OLNORDIORD8;
alter tablespace is_order_031 bufferpool OLNORDIORD8;
alter tablespace is_order_032 bufferpool OLNORDIORD8;

alter tablespace ts_history_001 bufferpool HST1;
alter tablespace ts_history_002 bufferpool HST1;
alter tablespace ts_history_003 bufferpool HST1;
alter tablespace ts_history_004 bufferpool HST1;
alter tablespace ts_history_005 bufferpool HST2;
alter tablespace ts_history_006 bufferpool HST2;
alter tablespace ts_history_007 bufferpool HST2;
alter tablespace ts_history_008 bufferpool HST2;
alter tablespace ts_history_009 bufferpool HST3;
alter tablespace ts_history_010 bufferpool HST3;
alter tablespace ts_history_011 bufferpool HST3;
alter tablespace ts_history_012 bufferpool HST3;
alter tablespace ts_history_013 bufferpool HST4;
alter tablespace ts_history_014 bufferpool HST4;
alter tablespace ts_history_015 bufferpool HST4;
alter tablespace ts_history_016 bufferpool HST4;
alter tablespace ts_history_017 bufferpool HST5;
alter tablespace ts_history_018 bufferpool HST5;
alter tablespace ts_history_019 bufferpool HST5;
alter tablespace ts_history_020 bufferpool HST5;
alter tablespace ts_history_021 bufferpool HST6;
alter tablespace ts_history_022 bufferpool HST6;
alter tablespace ts_history_023 bufferpool HST6;
alter tablespace ts_history_024 bufferpool HST6;
alter tablespace ts_history_025 bufferpool HST7;
alter tablespace ts_history_026 bufferpool HST7;
alter tablespace ts_history_027 bufferpool HST7;
alter tablespace ts_history_028 bufferpool HST7;
alter tablespace ts_history_029 bufferpool HST8;
alter tablespace ts_history_030 bufferpool HST8;
alter tablespace ts_history_031 bufferpool HST8;
alter tablespace ts_history_032 bufferpool HST8;

alter tablespace ts_newordA_001 bufferpool NEW1;
alter tablespace ts_newordA_002 bufferpool NEW1;
alter tablespace ts_newordA_003 bufferpool NEW1;
alter tablespace ts_newordA_004 bufferpool NEW1;
alter tablespace ts_newordA_005 bufferpool NEW2;
alter tablespace ts_newordA_006 bufferpool NEW2;
alter tablespace ts_newordA_007 bufferpool NEW2;
alter tablespace ts_newordA_008 bufferpool NEW2;
alter tablespace ts_newordA_009 bufferpool NEW3;
alter tablespace ts_newordA_010 bufferpool NEW3;
alter tablespace ts_newordA_011 bufferpool NEW3;
alter tablespace ts_newordA_012 bufferpool NEW3;
alter tablespace ts_newordA_013 bufferpool NEW4;
alter tablespace ts_newordA_014 bufferpool NEW4;
alter tablespace ts_newordA_015 bufferpool NEW4;
alter tablespace ts_newordA_016 bufferpool NEW4;
alter tablespace ts_newordA_017 bufferpool NEW5;
alter tablespace ts_newordA_018 bufferpool NEW5;
alter tablespace ts_newordA_019 bufferpool NEW5;
alter tablespace ts_newordA_020 bufferpool NEW5;
alter tablespace ts_newordA_021 bufferpool NEW6;
alter tablespace ts_newordA_022 bufferpool NEW6;
alter tablespace ts_newordA_023 bufferpool NEW6;
alter tablespace ts_newordA_024 bufferpool NEW6;
alter tablespace ts_newordA_025 bufferpool NEW7;
alter tablespace ts_newordA_026 bufferpool NEW7;
alter tablespace ts_newordA_027 bufferpool NEW7;
alter tablespace ts_newordA_028 bufferpool NEW7;
alter tablespace ts_newordA_029 bufferpool NEW8;
alter tablespace ts_newordA_030 bufferpool NEW8;
alter tablespace ts_newordA_031 bufferpool NEW8;
alter tablespace ts_newordA_032 bufferpool NEW8;

alter tablespace ts_newordB_001 bufferpool NEW1;
alter tablespace ts_newordB_002 bufferpool NEW1;
alter tablespace ts_newordB_003 bufferpool NEW1;
alter tablespace ts_newordB_004 bufferpool NEW1;
alter tablespace ts_newordB_005 bufferpool NEW2;
alter tablespace ts_newordB_006 bufferpool NEW2;
alter tablespace ts_newordB_007 bufferpool NEW2;
alter tablespace ts_newordB_008 bufferpool NEW2;
alter tablespace ts_newordB_009 bufferpool NEW3;
alter tablespace ts_newordB_010 bufferpool NEW3;
alter tablespace ts_newordB_011 bufferpool NEW3;
alter tablespace ts_newordB_012 bufferpool NEW3;
alter tablespace ts_newordB_013 bufferpool NEW4;
alter tablespace ts_newordB_014 bufferpool NEW4;
alter tablespace ts_newordB_015 bufferpool NEW4;
alter tablespace ts_newordB_016 bufferpool NEW4;
alter tablespace ts_newordB_017 bufferpool NEW5;
alter tablespace ts_newordB_018 bufferpool NEW5;
alter tablespace ts_newordB_019 bufferpool NEW5;
alter tablespace ts_newordB_020 bufferpool NEW5;
alter tablespace ts_newordB_021 bufferpool NEW6;
alter tablespace ts_newordB_022 bufferpool NEW6;
alter tablespace ts_newordB_023 bufferpool NEW6;
alter tablespace ts_newordB_024 bufferpool NEW6;
alter tablespace ts_newordB_025 bufferpool NEW7;
alter tablespace ts_newordB_026 bufferpool NEW7;
alter tablespace ts_newordB_027 bufferpool NEW7;
alter tablespace ts_newordB_028 bufferpool NEW7;
alter tablespace ts_newordB_029 bufferpool NEW8;
alter tablespace ts_newordB_030 bufferpool NEW8;
alter tablespace ts_newordB_031 bufferpool NEW8;
alter tablespace ts_newordB_032 bufferpool NEW8;

-- Size default bufferpools appropriately
alter bufferpool IBMDEFAULTBP  size 2000;
alter bufferpool IBMDEFAULT8K  size 200;
alter bufferpool IBMDEFAULT16K size 200;

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' 256
             )
       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' 256
             )
       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' 256
             )
       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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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' 256
               )
        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/raw41' 768
               )
        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/raw42' 768
               )
        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/raw43' 768
               )
        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/raw44' 768
               )
        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/raw45' 768
               )
        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/raw46' 768
               )
        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/raw47' 768
               )
        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/raw48' 768
               )
        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/raw49' 768
               )
```

```
        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/raw50' 768
               )
        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/raw51' 768
               )
        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/raw52' 768
               )
        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/raw53' 768
               )
        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/raw54' 768
               )
        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/raw55' 768
               )
        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/raw56' 768
               )
        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/raw57' 768
               )
        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/raw58' 768
               )
        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/raw59' 768
               )
        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/raw60' 768
               )
        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/raw61' 768
               )
        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/raw62' 768
               )
        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/raw63' 768
               )
        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/raw64' 768
               )
        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/raw65' 768
               )
        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/raw66' 768
               )
        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/raw67' 768
               )
        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/raw68' 768
               )
```

```
        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/raw69' 768
               )
        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/raw70' 768
               )
        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/raw71' 768
               )
        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/raw72' 768
               )
        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/raw81' 64,
                device '/dev/raw/raw82' 64,
                device '/dev/raw/raw83' 64,
                device '/dev/raw/raw84' 64,
                device '/dev/raw/raw85' 64,
                device '/dev/raw/raw86' 64,
                device '/dev/raw/raw87' 64,
                device '/dev/raw/raw88' 64,
                device '/dev/raw/raw89' 64,
                device '/dev/raw/raw90' 64,
                device '/dev/raw/raw91' 64,
                device '/dev/raw/raw92' 64,
                device '/dev/raw/raw93' 64,
                device '/dev/raw/raw94' 64,
                device '/dev/raw/raw95' 64,
                device '/dev/raw/raw96' 64,
                device '/dev/raw/raw97' 64,
                device '/dev/raw/raw98' 64,
                device '/dev/raw/raw99' 64,
                device '/dev/raw/raw100' 64,
                device '/dev/raw/raw101' 64,
                device '/dev/raw/raw102' 64,
                device '/dev/raw/raw103' 64,
                device '/dev/raw/raw104' 64,
                device '/dev/raw/raw105' 64,
                device '/dev/raw/raw106' 64,
                device '/dev/raw/raw107' 64,
                device '/dev/raw/raw108' 64,
                device '/dev/raw/raw109' 64,
                device '/dev/raw/raw110' 64,
                device '/dev/raw/raw111' 64,
                device '/dev/raw/raw112' 64
               )
        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/raw121' 11490304
               )
        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/raw122' 11490304
               )
        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/raw123' 11490304
               )
        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/raw124' 11490304
               )
        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/raw125' 11490304
               )
        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/raw126' 11490304
               )
        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/raw127' 11490304
               )
        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/raw128' 11490304
               )
        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/raw129' 11490304
               )
        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/raw130' 11490304
               )
        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/raw131' 11490304
               )
```

```
        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/raw132' 11490304
               )
        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/raw133' 11490304
               )
        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/raw134' 11490304
               )
        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/raw135' 11490304
               )
        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/raw136' 11490304
               )
        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/raw137' 11490304
               )
        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/raw138' 11490304
               )
        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/raw139' 11490304
               )
        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/raw140' 11490304
               )
        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/raw141' 11490304
               )
        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/raw142' 11490304
               )
        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/raw143' 11490304
               )
        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/raw144' 11490304
               )
        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/raw145' 11490304
               )
        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/raw146' 11490304
               )
        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/raw147' 11490304
               )
        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/raw148' 11490304
               )
        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/raw149' 11490304
               )
        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/raw150' 11490304
               )
```

```
        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/raw151' 11490304
               )
        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/raw152' 11490304
               )
        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/raw161' 8273408
               )
        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/raw162' 8273408
               )
        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/raw163' 8273408
               )
        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/raw164' 8273408
               )
        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/raw165' 8273408
               )
        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/raw166' 8273408
               )
        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/raw167' 8273408
               )
        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/raw168' 8273408
               )
        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/raw169' 8273408
               )
        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/raw170' 8273408
               )
        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/raw171' 8273408
               )
        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/raw172' 8273408
               )
        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/raw173' 8273408
               )
        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/raw174' 8273408
               )
        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/raw175' 8273408
               )
        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/raw176' 8273408
               )
        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/raw177' 8273408
```

```
            )
        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/raw178' 8273408
               )
        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/raw179' 8273408
               )
        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/raw180' 8273408
               )
        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/raw181' 8273408
               )
        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/raw182' 8273408
               )
        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/raw183' 8273408
               )
        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/raw184' 8273408
               )
        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/raw185' 8273408
               )
        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/raw186' 8273408
               )
        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/raw187' 8273408
               )
        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/raw188' 8273408
               )
        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/raw189' 8273408
               )
        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/raw190' 8273408
               )
        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/raw191' 8273408
               )
        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/raw192' 8273408
               )
        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/raw201' 262400
               )
        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/raw202' 262400
               )
        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/raw203' 262400
               )
        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/raw204' 262400
               )
        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/raw205' 262400
               )
        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/raw206' 262400
               )
        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/raw207' 262400
               )
        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/raw208' 262400
               )
        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/raw209' 262400
               )
        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/raw210' 262400
               )
        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/raw211' 262400
               )
        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/raw212' 262400
               )
        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/raw213' 262400
               )
        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/raw214' 262400
               )
        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/raw215' 262400
               )
        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/raw216' 262400
               )
        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/raw217' 262400
               )
        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/raw218' 262400
               )
        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/raw219' 262400
               )
        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/raw220' 262400
               )
        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/raw221' 262400
               )
        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/raw222' 262400
               )
        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/raw223' 262400
               )
        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/raw224' 262400
               )
        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/raw225' 262400
               )
        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/raw226' 262400
               )
        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/raw227' 262400
               )
        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/raw228' 262400
               )
        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/raw229' 262400
               )
        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/raw230' 262400
               )
        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/raw231' 262400
               )
        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/raw232' 262400
               )
        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/raw241' 198656
               )
        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/raw242' 198656
               )
        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/raw243' 198656
               )
        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/raw244' 198656
               )
        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/raw245' 198656
               )
        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/raw246' 198656
               )
        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/raw247' 198656
               )
        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/raw248' 198656
               )
        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/raw249' 198656
               )
        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/raw250' 198656
               )
        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/raw251' 198656
               )
        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/raw252' 198656
               )
        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/raw253' 198656
               )
        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/raw254' 198656
               )
        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/raw255' 198656
               )
        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/raw256' 198656
               )
        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/raw257' 198656
               )
        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/raw258' 198656
               )
        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/raw259' 198656
               )
        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/raw260' 198656
               )
        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/raw261' 198656
               )
        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/raw262' 198656
               )
        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/raw263' 198656
               )
        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/raw264' 198656
               )
        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/raw265' 198656
               )
        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/raw266' 198656
               )
        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/raw267' 198656
               )
        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/raw268' 198656
               )
        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/raw269' 198656
               )
        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/raw270' 198656
               )
        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/raw271' 198656
               )
        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/raw272' 198656
               )
        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/raw281' 252416
               )
        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/raw282' 252416
               )
        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/raw283' 252416
               )
        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/raw284' 252416
               )
        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/raw285' 252416
               )
        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/raw286' 252416
               )
        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/raw287' 252416
               )
        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/raw288' 252416
               )
        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/raw289' 252416
               )
        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/raw290' 252416
               )
        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/raw291' 252416
               )
        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/raw292' 252416
               )
        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/raw293' 252416
               )
        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/raw294' 252416
               )
        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/raw295' 252416
               )
        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/raw296' 252416
               )
        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/raw297' 252416
               )
        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/raw298' 252416
               )
        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/raw299' 252416
               )
        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/raw300' 252416
               )
        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/raw301' 252416
               )
        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/raw302' 252416
               )
        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/raw303' 252416
               )
        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/raw304' 252416
               )
        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/raw305' 252416
               )
        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/raw306' 252416
               )
        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/raw307' 252416
               )
        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/raw308' 252416
               )
        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/raw309' 252416
               )
        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/raw310' 252416
               )
        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/raw311' 252416
               )
        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/raw312' 252416
               )
        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/raw321' 217600
               )
        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/raw322' 217600
               )
        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/raw323' 217600
             )
       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/raw324' 217600
             )
       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/raw325' 217600
             )
       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/raw326' 217600
             )
       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/raw327' 217600
             )
       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/raw328' 217600
             )
       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/raw329' 217600
             )
       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/raw330' 217600
             )
       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/raw331' 217600
             )
       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/raw332' 217600
             )
       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/raw333' 217600
             )
       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/raw334' 217600
             )
       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/raw335' 217600
             )
       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/raw336' 217600
             )
       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/raw337' 217600
             )
       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/raw338' 217600
             )
       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/raw339' 217600
             )
       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/raw340' 217600
             )
       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/raw341' 217600
             )
       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/raw342' 217600
       )
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/raw343' 217600
       )
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/raw344' 217600
       )
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/raw345' 217600
       )
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/raw346' 217600
       )
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/raw347' 217600
       )
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/raw348' 217600
       )
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/raw349' 217600
       )
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/raw350' 217600
       )
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/raw351' 217600
       )
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/raw352' 217600
       )
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/raw361' 6587904
       )
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/raw362' 6587904
       )
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/raw363' 6587904
       )
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/raw364' 6587904
       )
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/raw365' 6587904
       )
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/raw366' 6587904
       )
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/raw367' 6587904
       )
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/raw368' 6587904
       )
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/raw369' 6587904
             )
       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/raw370' 6587904
             )
       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/raw371' 6587904
             )
       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/raw372' 6587904
             )
       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/raw373' 6587904
             )
       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/raw374' 6587904
             )
       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/raw375' 6587904
             )
       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/raw376' 6587904
             )
       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/raw377' 6587904
             )
       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/raw378' 6587904
             )
       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/raw379' 6587904
             )
       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/raw380' 6587904
             )
       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/raw381' 6587904
             )
       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/raw382' 6587904
             )
       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/raw383' 6587904
             )
       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/raw384' 6587904
             )
       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/raw385' 6587904
             )
       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/raw386' 6587904
             )
       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/raw387' 6587904
             )
       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/raw388' 6587904
             )
       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/raw389' 6587904
             )
       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/raw390' 6587904
             )
       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/raw391' 6587904
             )
       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/raw392' 6587904
             )
       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/raw401' 124928
             )
       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/raw402' 124928
             )
       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/raw403' 124928
             )
       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/raw404' 124928
             )
       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/raw405' 124928
             )
       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/raw406' 124928
             )
       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/raw407' 124928
             )
       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/raw408' 124928
             )
       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/raw409' 124928
             )
       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/raw410' 124928
             )
       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/raw411' 124928
             )
       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/raw412' 124928
             )
       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/raw413' 124928
             )
       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/raw414' 124928
             )
       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/raw415' 124928
             )
       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/raw416' 124928
             )
       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/raw417' 124928
             )
       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/raw418' 124928
             )
       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/raw419' 124928
             )
       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/raw420' 124928
             )
       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/raw421' 124928
             )
       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/raw422' 124928
             )
       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/raw423' 124928
             )
       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/raw424' 124928
             )
       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/raw425' 124928
             )
       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/raw426' 124928
             )
       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/raw427' 124928
             )
       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/raw428' 124928
             )
       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/raw429' 124928
             )
       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/raw430' 124928
             )
       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/raw431' 124928
             )
       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/raw432' 124928
             )
       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/raw441' 124928
             )
       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/raw442' 124928
               )
       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/raw443' 124928
               )
       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/raw444' 124928
               )
       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/raw445' 124928
               )
       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/raw446' 124928
               )
       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/raw447' 124928
               )
       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/raw448' 124928
               )
       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/raw449' 124928
               )
       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/raw450' 124928
               )
       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/raw451' 124928
               )
       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/raw452' 124928
               )
       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/raw453' 124928
               )
       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/raw454' 124928
               )
       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/raw455' 124928
               )
       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/raw456' 124928
               )
       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/raw457' 124928
               )
       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/raw458' 124928
               )
       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/raw459' 124928
               )
       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/raw460' 124928
               )
       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/raw461' 124928
             )
       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/raw462' 124928
             )
       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/raw463' 124928
             )
       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/raw464' 124928
             )
       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/raw465' 124928
             )
       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/raw466' 124928
             )
       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/raw467' 124928
             )
       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/raw468' 124928
             )
       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/raw469' 124928
             )
       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/raw470' 124928
             )
       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/raw471' 124928
             )
       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/raw472' 124928
             )
       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 1800;

alter bufferpool WDS1         size 4000;
alter bufferpool WDS2         size 4000;
alter bufferpool WDS3         size 4000;
alter bufferpool WDS4         size 4000;
alter bufferpool WDS5         size 4000;
alter bufferpool WDS6         size 4000;
alter bufferpool WDS7         size 4000;
alter bufferpool WDS8         size 4000;

alter bufferpool CST1         size 4032;
alter bufferpool CST2         size 4032;
alter bufferpool CST3         size 4032;
alter bufferpool CST4         size 4032;
alter bufferpool CST5         size 4032;
alter bufferpool CST6         size 4032;
alter bufferpool CST7         size 4032;
alter bufferpool CST8         size 4032;

alter bufferpool NEW1         size 152000;
alter bufferpool NEW2         size 152000;
alter bufferpool NEW3         size 152000;
alter bufferpool NEW4         size 152000;
alter bufferpool NEW5         size 152000;
alter bufferpool NEW6         size 152000;
alter bufferpool NEW7         size 152000;
alter bufferpool NEW8         size 152000;

alter bufferpool HST1         size 8000;
alter bufferpool HST2         size 8000;
alter bufferpool HST3         size 8000;
alter bufferpool HST4         size 8000;
alter bufferpool HST5         size 8000;
alter bufferpool HST6         size 8000;
alter bufferpool HST7         size 8000;
alter bufferpool HST8         size 8000;

alter bufferpool CSTI1        size 68000;
alter bufferpool CSTI2        size 68000;
alter bufferpool CSTI3        size 68000;
alter bufferpool CSTI4        size 68000;
alter bufferpool CSTI5        size 68000;
alter bufferpool CSTI6        size 68000;
alter bufferpool CSTI7        size 68000;
alter bufferpool CSTI8        size 68000;

alter bufferpool OLNORDIORD1 size 360000;
alter bufferpool OLNORDIORD2 size 360000;
alter bufferpool OLNORDIORD3 size 360000;
alter bufferpool OLNORDIORD4 size 360000;
alter bufferpool OLNORDIORD5 size 360000;
alter bufferpool OLNORDIORD6 size 360000;
alter bufferpool OLNORDIORD7 size 360000;
alter bufferpool OLNORDIORD8 size 360000;

alter bufferpool STK1         size 6710000;
alter bufferpool STK2         size 6710000;
alter bufferpool STK3         size 6710000;
alter bufferpool STK4         size 6710000;
alter bufferpool STK5         size 6710000;
alter bufferpool STK6         size 6710000;
alter bufferpool STK7         size 6710000;
alter bufferpool STK8         size 6710000;


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|%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|%s|%d|%d|%d|%.2f|%.2f|%d\n
";
char fmtHist[]    = "%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 42016
#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

Home > Categories > Wired Networking > Network Cables > AMC (ARROWMICRO) > Item#:N82E16812105305

AMC CC5E-B14B 14 ft. Cat 5E Blue Cat 5E Blue Cable - OEM

CUSTOMER REVIEWS | SPECIFICATIONS

Image Viewer

$2.99

3 Business Day Shipping $4.99
(Not available in HI, AK and PR)

In Stock

ADD TO CART
ADD TO WISH LIST
EMAIL THIS PAGE
PRINT THIS PAGE
PRICE ALERT

Model	
Brand	AMC
Model	CC5E-B14B
Spec	
Cat	5E
Length	14 ft.
Color	Blue

Quantity Pricing

Units	Price	Shipping
1	$2.99	$4.99
2+	$2.39	$3.47 ea.
5+	$2.29	$2.43 ea.
10+	$1.99	$2.09 ea.

Special Offers

No Payments for 6 Months on purchases over $500 with your Newegg.com Preferred Account!
Click here for important disclosures

No Payments for 90 Days On purchases over $250with Bill Me Later®!
Click here for important disclosures

Similar Items

Not the product you're looking for? We can make some suggestions to help you decide on a product that fits your needs.
Click here to view similar products

MANUFACTURER INFO | RETURNS & REBATES

Manufacturer Warranty

Beyond any applicable Newegg return policy, this item is warranted independently by the product's Manufacturer. Below is a summary provided for convenience only and may not be accurate or current. Click here for full details.

Manufacturer Contact Info

- Manufacturer Product Page

ORDER TRACKING | CLEARANCE | SHOPPING CART

SEARCH: PHONE ORDERS: 800.287.2323

HOME | ABOUT US | PRODUCT RETURN | CUSTOMER SERVICE | CONTACT US | PHONE ORDERS | REBATES

Now accepting PayPal

Boston Acoustics Receptor Radio in Charcoal ... **$149**
Garmin Forerunner 305 GPS Personal Trainer - 010-00467-00 ... **$199.99**
SanDisk Sansa m230 512MB Blue MP3 Player SDMX3-512-A18 / SDMX3512A18 ... **$26**

Join to receive our latest specials

Select a Brand

- AUDIO
- AUTO ELECTRONICS
- BABY & KIDS
- CABLES
- CAMCORDERS
- CAMERAS
- CASES
- CELLULAR
- COMPUTER SYSTEM
- CONTROLLERS
- DVD & CD
- FAX MACHINES
- GAMING
- GPS NAVIGATION
- HEADSETS
- HEALTH & BEAUTY
- HOME & GARDEN
- INPUT DEVICES
- MEMORY
- MODEMS
- MONITORS
- MOTHERBOARDS
- MP3
- NETWORKING
- PDA HANDHELDS
- POWER
- PRINTERS
- PROCESSORS
- PROJECTORS

Your search for *"DGS1024D"* returned 1 products.

Top Search Matches:

D Link DGS 1024D 24 Port 10 100 1000 Unmanaged Rackmountable Desktop Switch	202.99

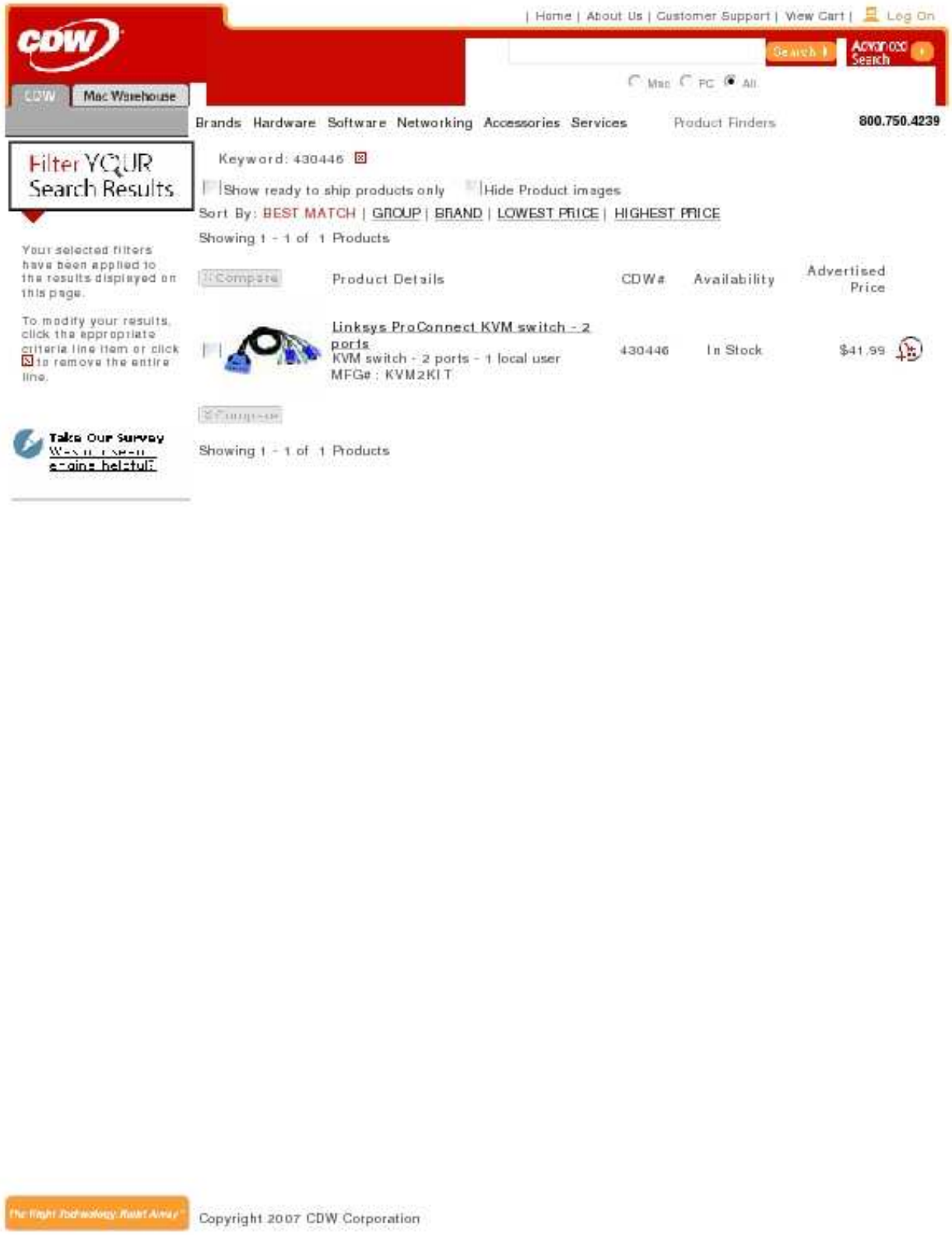
Check out our Closeout section for HOT specials on comparable products!

tell a friend

BizRate Smart Choice | CNET Certified Merchant | Shopping.com Certified Store | PriceGrabber.com click to see details | GeoTrust HACKER SAFE | mySimon Certified Store

HOME | ABOUT US | PRICE LIST | TECH SUPPORT | CUSTOMER SERVICE

CLOSEOUTS | REBATES | EXPRESS ORDER | PRIVACY STATEMENT


| 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
$41.99
Compare
Take Our Survey
Showing 1 - 1 of 1 Products
Copyright 2007 CDW Corporation

Jigantic Storefront

powered by PriceGrabber Storefronts

« Return to Price Comparison PriceGrabber.com

New Visual C++ Professional 6.0 Win32 Full Product CD-ROM

Manufacturer: Microsoft - Part Number: 04800210

Purchase Information

Price: $199.99
Availability: 4 in-stock
Payment Options: PayPal (Credit Cards)
Shipping Options / Cost: Estimate Shipping Below

Buy Now

Seller Information

Jigantic
4.63
(Read 72 Reviews)

Ask Seller a Question | See Seller's Inventory

Your order is secure up to $5.0.
Click here for more details.

Product Photos

*Important: This image is provided by PriceGrabber. PriceGrabber is not responsible for any discrepencies between the image and the item you are about to purchase.

Product Description

Brand New Factory Sealed Retail Packaging

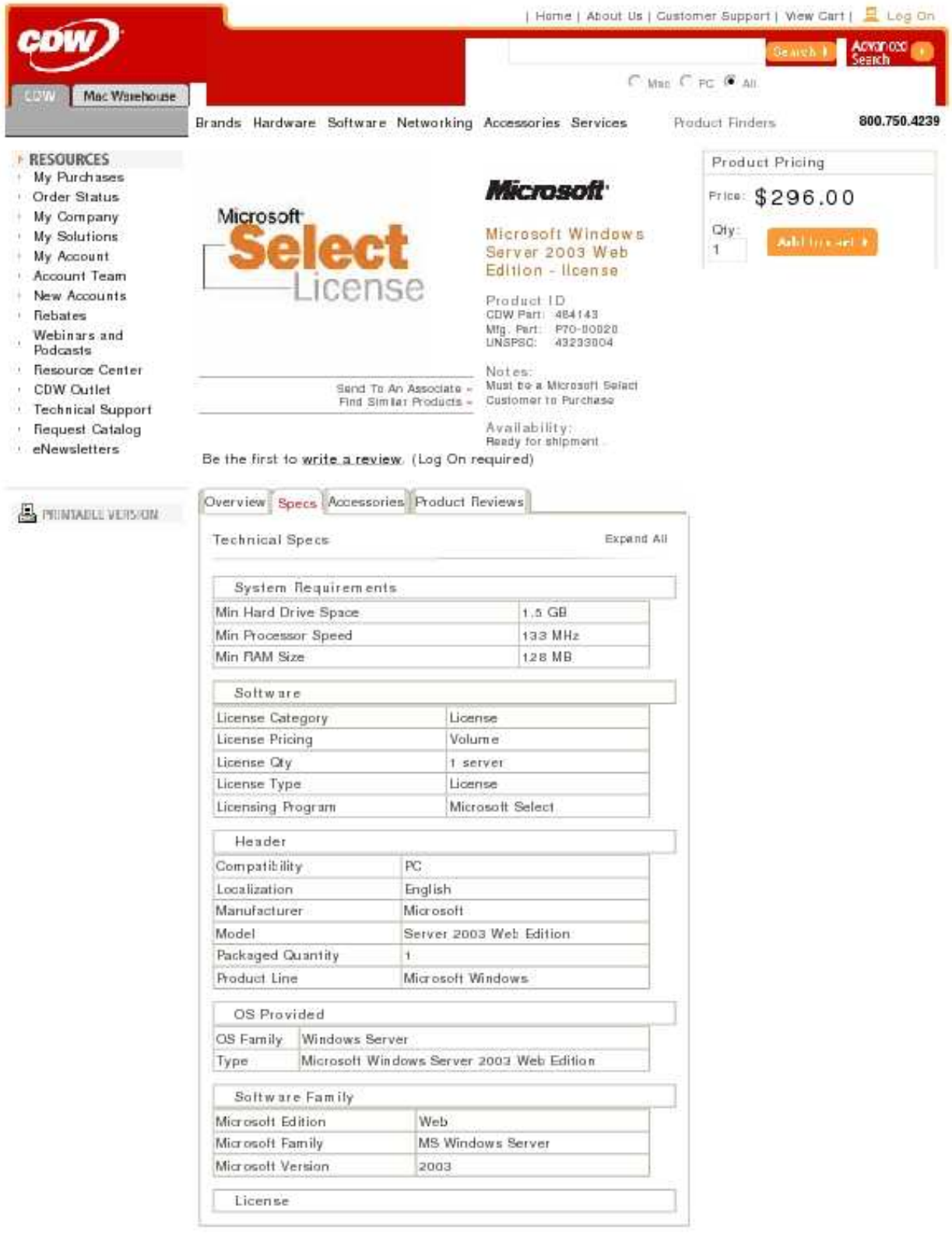
Shipping Details

Enter Zip Code to Estimate Shipping Costs: Go

Questions & Answers

There have been no questions posted thus far. Click here to ask this seller a question about this product.

To be notified of any responses, please make sure to login to your account before you ask or answer a question. If you do not have an account click here to register. Ask Seller a Question


| 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
Webinars and Podcasts
Resource Center
CDW Outlet
Technical Support
Request Catalog
eNewsletters
Microsoft Select License
Microsoft
Microsoft Windows Server 2003 Web Edition - license
Product ID
CDW Part: 484143
Mfg. Part: P70-00020
UNSPSC: 43233004
Notes:
Must be a Microsoft Select Customer to Purchase
Availability:
Ready for shipment
Send To An Associate »
Find Similar Products »
Be the first to write a review. (Log On required)
Product Pricing
Price: $296.00
Qty: 1
Add to cart
PRINTABLE VERSION
Overview Specs Accessories Product Reviews
Technical Specs
Expand All
System Requirements
Min Hard Drive Space 1.5 GB
Min Processor Speed 133 MHz
Min RAM Size 128 MB
Software
License Category License
License Pricing Volume
License Qty 1 server
License Type License
Licensing Program Microsoft Select
Header
Compatibility PC
Localization English
Manufacturer Microsoft
Model Server 2003 Web Edition
Packaged Quantity 1
Product Line Microsoft Windows
OS Provided
OS Family Windows Server
Type Microsoft Windows Server 2003 Web Edition
Software Family
Microsoft Edition Web
Microsoft Family MS Windows Server
Microsoft Version 2003
License

Cart | Register | Log In

redhat

- Home
- Solutions
- Services & Products
- Partners
- Developers
- Training
- Support
- Store

- Order History
- View Saved Carts
- Renew Subscriptions

Shopping Cart

Item	Quantity		Price	Line Total
New Subscription Contract October 8, 2007 - October 7, 2008				
Red Hat Enterprise Linux Advanced Platform, Premium (Unlimited Sockets)	1	Remove	$2,499.00	$2,499.00
Promotion Code:		Update Cart	Subtotal:	$2,499.00

Optional Install Discs and Documentation

add to cart Red Hat Enterprise Linux 5 (for Intel Itanium) $25 Media Kit (DVD only)
add to cart Red Hat Enterprise Linux 5 (for x86, AMD64, and Intel 64) $25 Media Kit (DVD only)

Continue shopping Continue to Checkout

ABOUT SSL CERTIFICATES

October 2, 2007

IBM Corporation
Mr. Ray Venditti
Linux Performance

Dear Ray:

The table shown below lists the U.S. pricing for DB2 9.5 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	278.52	222,816
SW Maintenance Renewal - 2 year	1600	13.27	21,232
		Sub-total reference price for DB2 ESE:	**244,048**
		TOTAL REFERENCE PRICE:	**244,048**

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