
IBM System x3950 M2

Using

DB2 9.5 Enterprise Edition

and

Red Hat Enterprise Linux Advanced Platform 5

TPC BenchmarkTM C

Full Disclosure Report



First Edition August 19, 2008

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

System Storage

ServeRAID

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.

Intel and Xeon are registered trademarks of Intel Corporation.

First Edition: August 19, 2008

The information contained in this document is distributed on an AS IS basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming, or services in your country.

All performance data contained in this publication was obtained in a controlled environment, and therefore the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data in their specific environment.

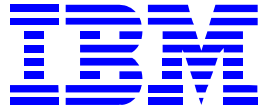
Request for additional copies of this document should be sent to the following address:

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

© Copyright International Business Machines Corporation, 2008 All rights reserved.

Permission is hereby granted to reproduce this document in whole or in part, provided the copyright notice printed above is set forth in full text on the title page of each item reproduced.

NOTE: US. Government Users - Documentation related to restricted rights: Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

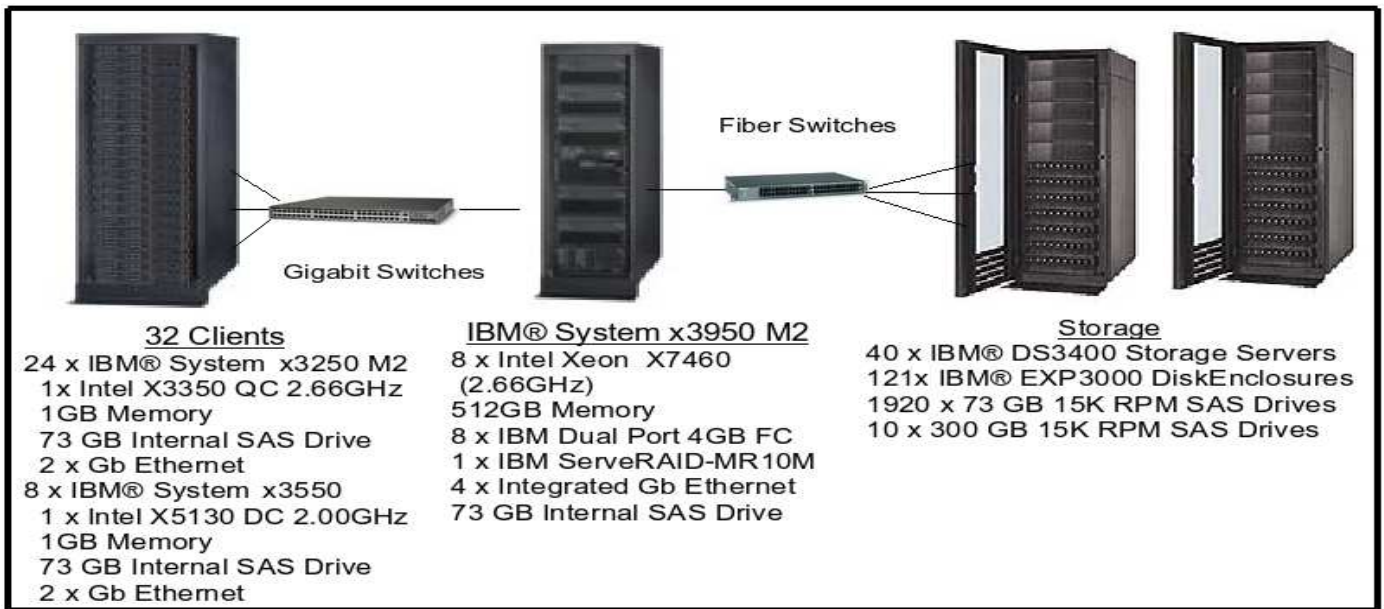


**IBM System x3950 M2
DB2® 9.5**

TPC-C Rev. 5.10

**Report Date:
August 19, 2008**

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
\$2,386,768 USD	1,200,632	\$1.99 USD	December 10, 2008	
Database Processors/Cores/Threads	Database Manager	Operating System	Other Software	No. Users
8/48/48 2.66 GHz Intel Xeon X7460	DB2 9.5	RHEL 5	Microsoft Visual C++ Microsoft COM+	960,000



System Components	Each of the 32 Clients		Server	
	Quantity	Description	Quantity	Description
Processors/Cores/Threads	1/4/4 1/2/2	2.66GHz 12MB L2 Xeon X3350 or 2.0GHz 4MB L2 Xeon X5130	8/48/48	2.66 GHz Intel Xeon X7460 3 x 3MB L2 1 x 16MB L3
Memory	1	1 GB	64	8 GB
Disk Controllers	1	SAS	1 1 8 40	Integrated SAS ServeRAID-MR10M 4Gb dual-port FC Adapters IBM DS3400 Controllers for Data
Disk Drives	1	73 GB	1921 10	73 GB 15K RPM SAS 300 GB 15K RPM SAS
Total Storage		2336 GB		143,233 GB
Terminals	1	System Console	1	System Console

IBM Corporation	IBM System x3950 M2 c/s				TPC-C Revision 5.10		
	with DB2 9.5				Report Date: August 19, 2008		
Description	Part Number	Brand	Price Source	Unit Price	Quantity	Extended Price	3-Yr. Maint. Price
Server Hardware							
x3950 M2 (2 x Intel Xeon Processor X7460 2.66GHz/9MB L2 Cache/16MB L3 Cache) with 4 memory cards and 8 x 1GB DIMM	72336SU	IBM	1a*	24,999	2	49,998	
Intel Xeon Processor X7460 (2.66GHz)	44E4473	IBM	1a*	7,999	4	31,996	
16GB (2x8GB) 667MHz PC2-5300 ECC DDR2 SDRAM DIMM	43V7356	IBM	1	6,999	32	223,968	
IBM 73.4GB 10K 2.5 inch SAS Drive	40K1052	IBM	1	259	1	259	
IBM T115 15-inch TFT Display	494215U	IBM	1a	209	1	209	
IBM Preferred Pro USB Keyboard	40K9584	IBM	1	29	1	29	
IBM 3-Button Optical Mouse - Black - USB	40K9201	IBM	1	19	1	19	
ServicePac for 3-Year 24x7x4 Support (x3950 M2)	10N3059	IBM	1	1,695	2		3,390
ServicePac for 3-Year 24x7x4 Support (Display)	30L9183	IBM	1	90	1		90
						306,478	3,480
Server Storage							
IBM 4-Gbps FC Dual-Port PCI-E HBA	39R6527	IBM	1	1,424	8	11,392	
ServeRAID-MR10M SAS/SATA Controller	43W4339	IBM	1	899	1	899	
IBM System Storage DS3400 Express	1726-42E	IBM	1	6,702	40	268,080	
IBM 3M SAS cable	39R6531	IBM	1	135	242	32,670	
IBM System Storage EXP3000	1727-01X	IBM	1	3,199	121	387,079	
IBM EXP3000 Environmental Services Module (ESM)	39R6515	IBM	1a	999	121	120,879	
IBM Hot-Swap 3.5 inch 73.4GB 15K SAS HDD	43W7523	IBM	1	329	1920	631,680	
300GB 15K 3.5" Hot-Swap SAS	43X0802	IBM	1	779	10	7,790	
IBM S2 42U Standard Rack	93074RX	IBM	1	1,489	8	11,912	
ServicePac for 3-Year 24x7x4 Support (DS3400)	44J8073	IBM	1	1,300	40		52,000
ServicePac for 3-Year 24x7x4 Support (EXP3000)	41L2768	IBM	1	760	121		91,960
ServicePac for 3-Year 24x7x4 Support (Rack)	41L2760	IBM	1	300	8		2,400
					Subtotal	1,472,381	146,360
Server Software							
DB2 ESE 9.5 SW License and Maintenance 12 Months		IBM	2	278.52	2400	668,448	
SW Maintenance Renewal - 1 Year		IBM	2	13.27	4800		63,696
Red Hat Enterprise Linux Advanced Platform, Premium (Unlimited Sockets)		Red Hat	3	6,747	1	6,747	
					Subtotal	675,195	63,696
Client Hardware							
IBM System x3550 with Microsoft Windows Server 2003 and 3yr 24x7x4 Maintenance & Warranty	7978MC1	IBM	1a	1,933	8	15,464	
2.0GHz Dual-Core X5130 2 x 512MB	1308	IBM	1a	64	8	512	
3.5" Hot Swap SAS Enabled System	2465	IBM	1a	598	8	4,784	
IBM System x3250 M2 with Microsoft Windows Server 2003 and 3yr 24x7x4 Maintenance & Warranty	4194MC1	IBM	1a - S	2,270	24	54,480	
2.66 Quad-Core X3350 2 x 512MB RAM	3652	IBM	1a -S	354	1	354	
					Subtotal	75,594	0
Client Software							
Visual Studio Standard 2005	127-00012	Microsoft	5	250	1	250	
Microsoft Problem Resolution Services		Microsoft	5	245	1		245
					Subtotal	250	245
Third Party Components							
Linksys ProConnect KVM Switch - 2 ports (2 spares)	430446	cdw	4	42	3	126	
D-LINK DGS-1024D 24-Port 10/100/1000 Switch (2 spares)	DGS1024D	cdw	4	210	8	1,680	
Qlogic SANbox 5600 16-Port 4Gb Fabric Switch (2 spares)	sb5600-16A	SANdirect	6	4,390	10	43,900	
UPS - powercom KIN-1500AP 1500VA 900Watts (2 spares)	N82E16842106115	newegg	7	135	3	405	
					Subtotal	46,111	0
					Total	2,576,009	213,781
IBM Large Purchase Discount (See Note 1.)	23.36%		1			403,022	
Pricing: 1 - IBM - 1-888-SHOP-IBM, ext. 5821; 2 - IBM; 3 -Red Hat; 4 - CDW.com; 5- Microsoft; 6 - SANdirect.com; 7 - newegg.com					Three-Year Cost of Ownership USD:		\$2,386,768
Note 1: Discount based on IBM Direct guidance applies to all line items where Pricing=1. Pricing is for this system or one of similar size.					tpmC:		1,200,632
* These components are not immediately orderable. See the FDR for more information.					\$ USD/tpmC:		\$1.99
S - One or more components of the measured configuration have been substituted in the Priced Configuration. See the FDR for details							
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 x3950 M2

MQTH, computed Maximum Qualified Throughput: 1,200,632 tpmC

<u>Response Times (in seconds)</u>	<u>90th %</u>	<u>Average</u>	<u>Maximum</u>
New Order	0.920	0.409	9.297
Payment	0.900	0.389	9.282
Order-Status	0.920	0.404	9.281
Delivery (interactive)	0.300	0.154	8.984
Delivery (deferred)	0.31	0.21	2.41
Stock-Level	1.020	0.484	9.407
Menu	0.282	0.153	9.468

Response time delay added for emulated components was 0.1 seconds

<u>Transaction Mix, in percent of total transactions</u>	<u>Percent</u>
New Order	44.951%
Payment	43.020%
Order-Status	4.010%
Delivery	4.009%
Stock-Level	4.009%

<u>Keying/Think Times (in seconds)</u>	<u>Min.</u>	<u>Average</u>	<u>Max.</u>
New Order	18.00/0.00	18.00/12.036	18.016/120.313
Payment	3.00/0.00	3.00/12.036	3.016/120.313
Order-Status	2.00/0.00	2.00/10.036	2.016/100.313
Delivery	2.00/0.00	2.00/5.038	2.016/50.313
Stock-Level	2.00/0.00	2.00/5.038	2.016/50.313

Test Duration

Ramp-up Time	18 minutes
Measurement interval	2 hours 10 minutes
Transactions during measurement interval (all types)	347,229,652

Checkpoints

Number of checkpoints	N/A
Checkpoint interval	N/A

Table of Contents

Preface.....	9
0 General Items	10
0.1. Application Code Disclosure	10
0.2. Benchmark Sponsor	10
0.3. Parameter Settings.....	10
0.4. Configuration Diagrams.....	10
1 Clause 1: Logical Data Base Design Related Items	13
1.1. Table Definitions.....	13
1.2. Database Organization	13
1.3. Insert and/or Delete Operations.....	13
1.4. Horizontal or Vertical Partitioning.....	13
2 Clause 2: Transaction & Terminal Profiles Related Items	14
2.1. Verification for the Random Number Generator.....	14
2.2. Input/Output Screens.....	14
2.3. Priced Terminal Features	14
2.4. Presentation Managers	14
2.5. Home and Remote Order-lines.....	14
2.6. New-Order Rollback Transactions.....	14
2.7. Number of Items per Order	14
2.8. Home and Remote Payment Transactions.....	14
2.9. Non-Primary Key Transactions.....	15
2.10. Skipped Delivery Transactions	15
2.11. Mix of Transaction Types	15
2.12. Queuing Mechanism of Delivery	15
3 Clause 3: Transaction and System Properties	17
3.1. Atomicity Requirements	17
3.2. Consistency Requirements	17
3.3. Isolation Requirements.....	18
3.4. Durability Requirements	18
4 Clause 4: Scaling and Data Base Population Related Items.....	20
4.1. Cardinality of Tables.....	20
4.2. Distribution of Tables and Logs.....	20
4.3. Data Base Model Implemented.....	21
4.4. Partitions/Replications Mapping	21
4.5. 60-Day Space Calculations	26
5 Clause 5: Performance Metrics and Response Time Related Items	27
5.1. Response Times	27
5.2. Keying and Think Times.....	27
5.3. Response Time Frequency Distribution	28
5.4. Performance Curve for Response Time versus Throughput	30
5.5. Think Time Frequency Distribution.....	31
5.6. Throughput versus Elapsed Time.....	32
5.7. Steady State Determination.....	32
5.8. Work Performed During Steady State.....	32
5.9. Measurement Interval.....	34
6 Clause 6: SUT, Driver, and Communication Definition Related Items	35
6.1. RTE Availability	35
6.2. Functionality and Performance of Emulated Components.....	35
6.3. Network Bandwidth	35
6.4. Operator Intervention	35
7 Clause 7: Pricing Related Items	36
7.1. Hardware and Programs Used.....	36
7.2. Three Year Cost of System Configuration.....	36
7.3. Availability Dates	36
7.4. Statement of tpmC and Price/Performance	37

8	Clause 9: Audit Related Items.....	38
9	Appendix A: Client Server Code.....	41
9.1.	Client/Terminal Handler Code.....	41
9.2.	Client/Terminal Handler Code.....	52
10	Appendix B: Tunable Parameters.....	85
10.1.	Database Parameters.....	85
10.2.	Transaction Monitor Parameters.....	87
10.3.	Linux Parameters.....	89
11	Appendix C: Database Setup Code.....	101
11.1.	Database Creation Scripts.....	101
11.2.	Data Generation.....	512
12	Appendix D: Pricing.....	524

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.10 dated April, 2008, for measurements on the IBM System x3950 M2. The software used on the IBM System x3950 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 x3950 M2

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM System x3950 M2	DB2 9.5	Red Hat Enterprise Linux Advanced Platform 5

Total System Cost	TPC-C Throughput	Price/Performance
<ul style="list-style-type: none">• Hardware• Software• 3 Years Maintenance	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC
\$2,386,768 USD	1,200,632	\$1.99 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 5.10 in April 2008.

This is the full disclosure report for benchmark testing of the IBM System x3950 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.*

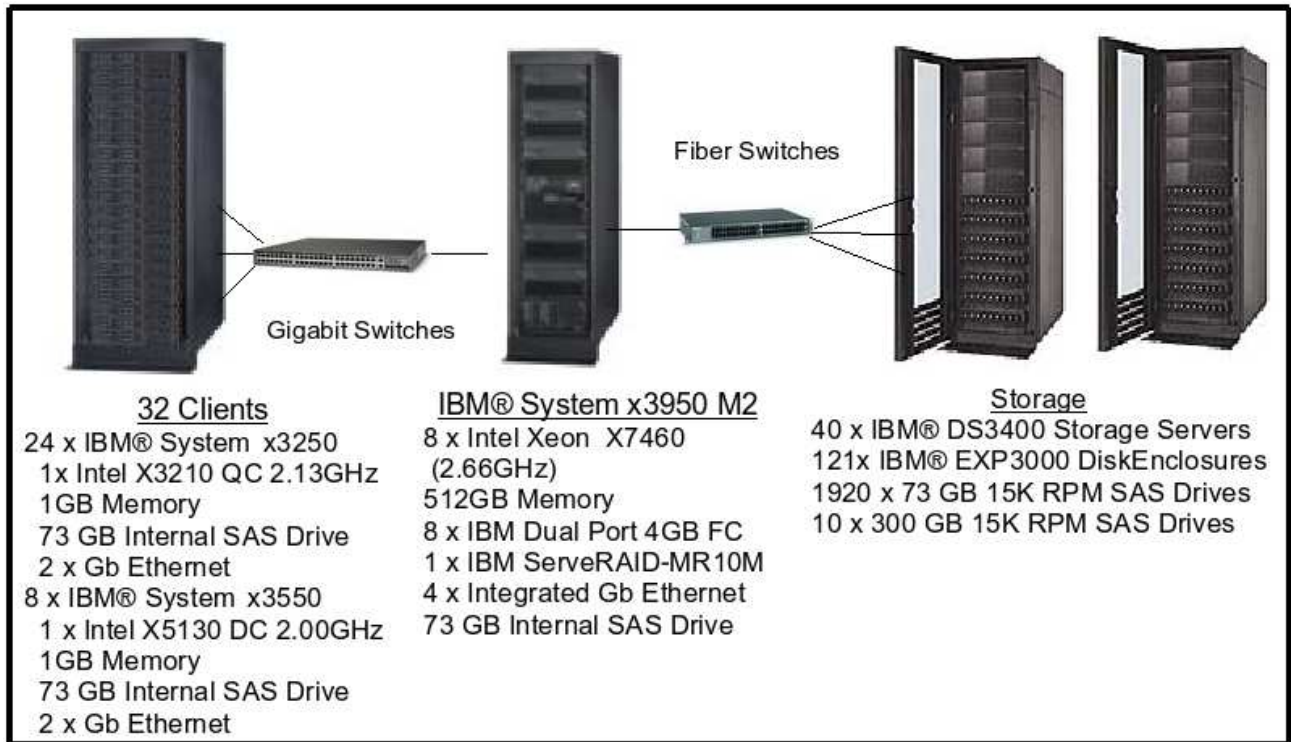
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

0.4. Configuration Diagrams

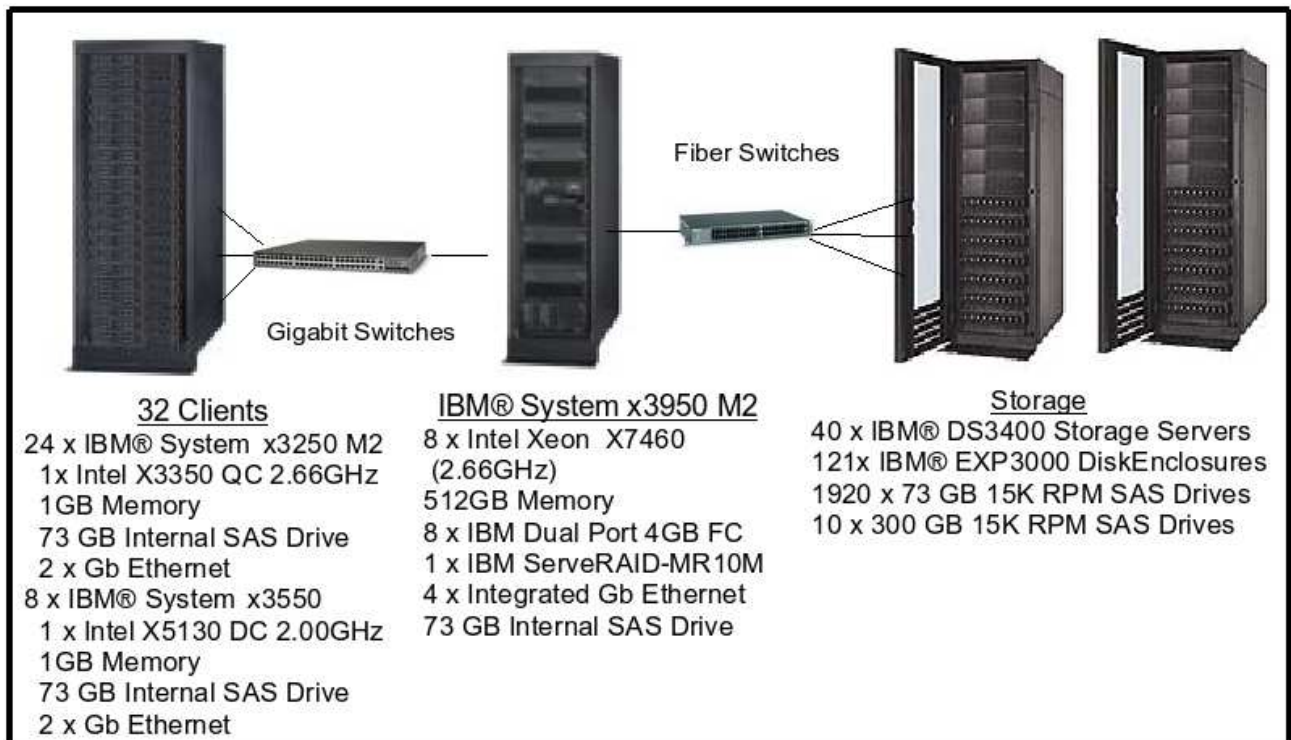
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *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 x3950 M2 Benchmark Configuration



IBM System x3950 M2 Priced Configuration



Twenty four x3250M2 clients were priced for the twenty four x3250 clients used in the benchmark. Both systems use quad core Intel Xeon chips with x3250M2 having the faster chip (2.66GHz vs 2.13GHz) and larger cache (12MB L2 vs. 8MB L2). The x3250M2 also has a faster bus speed (1333MHz vs 1066MHz). Both machines used the same SAS controller and SAS disk drive. The auditor reviewed the specifications of the 2 machines and certified that the pricing upgrade was compliant with the TPC-C specification.

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 1200 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).

Microsoft Internet Explorer was used to verify the compliance with clause 2.2.2.4.

2.4. Presentation Managers

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

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

2.5. Home and Remote Order-lines

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

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

2.6. New-Order Rollback Transactions

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

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

2.7. Number of Items per Order

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

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

2.8. Home and Remote Payment Transactions

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

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

2.9. Non-Primary Key Transactions

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

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

2.10. Skipped Delivery Transactions

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

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

2.11. Mix of Transaction Types

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

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

New Order	IBM System x3950 M2
Percentage of Home order lines	99.00%
Percentage of Remote order lines	1.00%
Percentage of Rolled Back Transactions	1.001%
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	59.999%
Percentage of Order-Status using C_LAST	59.990%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.951%
Payment	43.020%
Order-Status	4.010%
Delivery	4.009%
Stock-Level	4.009%

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 preceding 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:

$$\max(no_o_id) - \min(no_o_id) + 1 = \text{number of rows in New Order for the Warehouse/District}$$

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:

$$\sum(o_ol_cnt) = \text{number of rows in the Order Line table for the Warehouse/District}$$

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:

This test was conducted on a 12.5% 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 12.5% load test was started and allowed to run for over 5 minutes.
3. One of the disks containing the transaction log was removed. Since the log was implemented as a RAID-10 array, DB2 continued to process the transactions successfully.
4. The test continued for at least another 5 minutes.
5. The test was ended and the database was deactivated in a controller manner.

6. 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.
7. 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	96,000
District	960,000
Customer	2,880,000,000
History	2,880,000,000
Orders	2,880,000,000
New Order	864,000,000
Order Line	28,800,052,057
Stock	9,600,000,000
Item	100,000

Table 4-1: Initial Cardinality of Tables

4.2. Distribution of Tables and Logs

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

The log was configured using one ServeRAID-MR10M controller. This controller was connected to a RAID-10 disk array consisting of ten 300GB hot-swap SAS disks housed in one EXP3000 drive enclosure and backed by a UPS.

There are 80 Logical Disks (LD) for the database tables. There are 8 dual port storage adapters for the tables. Each dual port adapter is assigned 10 LDs. Each LD is configured as a RAID0 disk array, with 24 physical disks. There are a total of 1920 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. Partition are laid out on an LD as follows:

Partition#	Blocks	Container Usage
1	7968	Warehouse
2	8001	District
3	7969	Item
4		Extended partition
5	44957808	Stock
6	32933170	Customer
7	3212922	Customer Index
8	4016180	History
9	2409674	Orders
10	2409696	Order Index
11	53817718	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	80 tablespaces each using 1 LD
District	80 tablespaces each using 1 LD
Item	One tablespace using 80 LDs
Stock	80 tablespaces each using 1 LD
Customer	80 tablespaces each using 1 LD
Customer Index	80 tablespaces each using 1 LD
History	80 tablespaces each using 1 LD
Orders	80 tablespaces each using 1 LD
Order Index	80 tablespaces each using 1 LD
Order Line	80 tablespaces each using 1 LD
New OrdersA	80 tablespaces each using 1 LD
New OrdersB	80 tablespaces each using 1 LD

4.3. Data Base Model Implemented

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

The database manager used for this testing was DB2 9.5. 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.

4.4. Partitions/Replications Mapping

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

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

Data Distribution Logical Disks (LDs)			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD10 Partition 1	FC1	raw1 –raw10	ts_wh_01-ts_wh_10
LD11 – LD20 Partition 1	FC2	Raw11 –raw20	ts_wh_11-ts_wh_20
LD21 – LD30 Partition 1	FC3	Raw21 –raw30	ts_wh_21-ts_wh_30
LD31 – LD40 Partition 1	FC4	Raw31 –raw40	ts_wh_31-ts_wh_40
LD41 – LD50 Partition 1	FC5	Raw51 –raw50	ts_wh_41-ts_wh_50
LD51 – LD60 Partition 1	FC6	Raw51 –raw60	ts_wh_51-ts_wh_60
LD61 – LD70 Partition 1	FC7	Raw61 –raw70	ts_wh_61-ts_wh_70
LD71 – LD80 Partition 1	FC8	Raw71 –raw80	ts_wh_71-ts_wh_80
LD1 – LD10 Partition 2	FC1	Raw81 –raw90	ts_dis_01-ts_dis_10
LD11 – LD20 Partition 2	FC2	Raw91 –raw100	ts_dis_11-ts_dis_20
LD21 – LD30 Partition 2	FC3	Raw101 –raw110	ts_dis_21-ts_dis_30
LD31 – LD40 Partition 2	FC4	Raw111 –raw120	ts_dis_31-ts_dis_40
LD41 – LD50 Partition 2	FC5	Raw121 –raw130	ts_dis_41-ts_dis_50
LD51 – LD60 Partition 2	FC6	Raw131 –raw140	ts_dis_51-ts_dis_60
LD61 – LD70 Partition 2	FC7	Raw141 –raw150	ts_dis_61-ts_dis_70
LD71 – LD80 Partition 2	FC8	Raw151 –raw160	ts_dis_71-ts_dis_80
LD1 – LD10 Partition 3	FC1	Raw161 –raw170	ts_item
LD11 – LD20 Partition 3	FC2	Raw171 –raw180	ts_item
LD21 – LD30 Partition 3	FC3	Raw181 –raw190	ts_item
LD31 – LD40 Partition 3	FC4	Raw191 –raw200	ts_item
LD41 – LD50 Partition 3	FC5	Raw201 –raw210	ts_item
LD51 – LD60 Partition 3	FC6	Raw211 –raw220	ts_item
LD61 – LD70 Partition 3	FC7	Raw221 –raw230	ts_item
LD71 – LD80 Partition 3	FC8	Raw231 –raw240	ts_item

Table 4-2: IBM System x3950 M2 Data Distribution Benchmark Configuration

Data Distribution Logical Disks (LDs)			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD10 Partition 5	FC1	Raw241–raw250	ts_stock_01–ts_stock_10
LD11 – LD20 Partition 5	FC2	Raw251–raw260	ts_stock_11–ts_stock_20
LD21 – LD30 Partition 5	FC3	Raw261–raw270	ts_stock_21–ts_stock_30
LD31 – LD40 Partition 5	FC4	Raw271 –raw280	ts_stock_31–ts_stock_40
LD41 – LD50 Partition 5	FC5	Raw281 –raw290	ts_stock_41–ts_stock_50
LD51 – LD60 Partition 5	FC6	Raw291 –raw300	ts_stock_51–ts_stock_60
LD61 – LD70 Partition 5	FC7	Raw301 –raw310	ts_stock_61–ts_stock_70
LD71 – LD80 Partition 5	FC8	Raw311 –raw320	ts_stock_71–ts_stock_80
LD1 – LD10 Partition 6	FC1	Raw321 –raw330	ts_cust_01–ts_cust_10
LD11 – LD20 Partition 6	FC2	Raw33 –raw340	ts_cust_11–ts_cust_20
LD21 – LD30 Partition 6	FC3	Raw341 –raw350	ts_cust_21–ts_cust_30
LD31 – LD40 Partition 6	FC4	Raw351 –raw360	ts_cust_31–ts_cust_40
LD41 – LD50 Partition 6	FC5	Raw361 –raw370	ts_cust_41–ts_cust_50
LD51 – LD60 Partition 6	FC6	Raw371 –raw380	ts_cust_51–ts_cust_60
LD61 – LD70 Partition 6	FC7	Raw381 –raw390	ts_cust_61–ts_cust_70
LD71 – LD80 Partition 6	FC8	Raw391 –raw400	ts_cust_71–ts_cust_80
LD1 – LD10 Partition 7	FC1	Raw401 –raw410	is_cust_01–is_cust_10
LD11 – LD20 Partition 7	FC2	Raw411 –raw420	is_cust_11–is_cust_20
LD21 – LD30 Partition 7	FC3	Raw421 –raw430	is_cust_21–is_cust_30
LD31 – LD40 Partition 7	FC4	Raw431 –raw440	is_cust_31–is_cust_40
LD41 – LD50 Partition 7	FC5	Raw441 –raw450	is_cust_41–is_cust_50
LD51 – LD60 Partition 7	FC6	Raw451 –raw460	is_cust_51–is_cust_60
LD61 – LD70 Partition 7	FC7	Raw461 –raw470	is_cust_61–is_cust_70
LD71 – LD80 Partition 7	FC8	Raw471 –raw480	is_cust_71–is_cust_80

Table 4-3: IBM System x3950 M2 Data Distribution Benchmark Configuration

Data Distribution Logical Disks (LDs)			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD10 Partition 8	FC1	Raw481–raw490	ts_hist_01-ts_hist_10
LD11 – LD20 Partition 8	FC2	Raw491–raw500	ts_hist_11-ts_hist_20
LD21 – LD30 Partition 8	FC3	Raw501–raw510	ts_hist_21-ts_hist_30
LD31 – LD40 Partition 8	FC4	Raw511 –raw520	ts_hist_31-ts_hist_40
LD41 – LD50 Partition 8	FC5	Raw521 –raw530	ts_hist_41-ts_hist_50
LD51 – LD60 Partition 8	FC6	Raw531 –raw540	ts_hist_51-ts_hist_60
LD61 – LD70 Partition 8	FC7	Raw541 –raw550	ts_hist_61-ts_hist_70
LD71 – LD80 Partition 8	FC8	Raw551 –raw560	ts_hist_71-ts_hist_80
LD1 – LD10 Partition 9	FC1	Raw561 –raw570	ts_ord_01-ts_ord_10
LD11 – LD20 Partition9	FC2	Raw571 –raw580	ts_ord_11-ts_ord_20
LD21 – LD30 Partition 9	FC3	Raw581 –raw590	ts_ord_21-ts_ord_30
LD31 – LD40 Partition 9	FC4	Raw591 –raw600	ts_ord_31-ts_ord_40
LD41 – LD50 Partition 9	FC5	Raw601 –raw610	ts_ord_41-ts_ord_50
LD51 – LD60 Partition 9	FC6	Raw611 –raw620	ts_ord_51-ts_ord_60
LD61 – LD70 Partition 9	FC7	Raw621 –raw630	ts_ord_61-ts_ord_70
LD71 – LD80 Partition 9	FC8	Raw631 –raw640	ts_ord_71-ts_ord_80
LD1 – LD10 Partition 10	FC1	Raw641 –raw650	is_ord_01-is_ord_10
LD11 – LD20 Partition 10	FC2	Raw651 –raw660	is_ord_11-is_ord_20
LD21 – LD30 Partition 10	FC3	Raw661 –raw670	is_ord_21-is_ord_30
LD31 – LD40 Partition 10	FC4	Raw671 –raw680	is_ord_31-is_ord_40
LD41 – LD50 Partition 10	FC5	Raw681 –raw690	is_ord_41-is_ord_50
LD51 – LD60 Partition 10	FC6	Raw691 –raw700	is_ord_51-is_ord_60
LD61 – LD70 Partition 10	FC7	Raw701 –raw710	is_ord_61-is_ord_70
LD71 – LD80 Partition 10	FC8	Raw711 –raw720	is_ord_71-is_ord_80

Table 4-4: IBM System x3950 M2 Data Distribution Benchmark Configuration

Data Distribution Logical Disks (LDs)			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD10 Partition 11	FC1	Raw721–raw730	ts_orderline_01-ts_orderline_10
LD11 – LD20 Partition 11	FC2	Raw731–raw740	ts_orderline_11-ts_orderline_20
LD21 – LD30 Partition 11	FC3	Raw741–raw750	ts_orderline_21-ts_orderline_30
LD31 – LD40 Partition 11	FC4	Raw751 –raw760	ts_orderline_31-ts_orderline_40
LD41 – LD50 Partition 11	FC5	Raw761 –raw770	ts_orderline_41-ts_orderline_50
LD51 – LD60 Partition 11	FC6	Raw771 –raw780	ts_orderline_51-ts_orderline_60
LD61 – LD70 Partition 11	FC7	Raw781 –raw790	ts_orderline_61-ts_orderline_70
LD71 – LD80 Partition 11	FC8	Raw701 –raw800	ts_orderline_71-ts_orderline_80
LD1 – LD10 Partition 12	FC1	Raw801 –raw810	ts_newordA_01-ts_newordA_10
LD11 – LD20 Partition12	FC2	Raw811 –raw820	ts_newordA_11-ts_newordA_20
LD21 – LD30 Partition 12	FC3	Raw821 –raw830	ts_newordA_21-ts_newordA_30
LD31 – LD40 Partition 12	FC4	Raw831 –raw840	ts_newordA_31-ts_newordA_40
LD41 – LD50 Partition 12	FC5	Raw841 –raw850	ts_newordA_41-ts_newordA_50
LD51 – LD60 Partition 12	FC6	Raw851 –raw860	ts_newordA_51-ts_newordA_60
LD61 – LD70 Partition 12	FC7	Raw861 –raw870	ts_newordA_61-ts_newordA_70
LD71 – LD80 Partition 12	FC8	Raw871 –raw880	ts_newordA_71-ts_newordA_80
LD1 – LD10 Partition 13	FC1	Raw881 –raw890	ts_newordB_01-ts_newordB_10
LD11 – LD20 Partition 13	FC2	Raw891 –raw900	ts_newordB_11-ts_newordB_20
LD21 – LD30 Partition 13	FC3	Raw901 –raw910	ts_newordB_21-ts_newordB_30
LD31 – LD40 Partition 13	FC4	Raw911 –raw920	ts_newordB_31-ts_newordB_40
LD41 – LD50 Partition 13	FC5	Raw921 –raw930	ts_newordB_41-ts_newordB_50
LD51 – LD60 Partition 13	FC6	Raw931 –raw940	ts_newordB_51-ts_newordB_60
LD61 – LD70 Partition 13	FC7	Raw941 –raw950	ts_newordB_61-ts_newordB_70
LD71 – LD80 Partition 13	FC8	Raw951 –raw960	ts_newordB_71-ts_newordB_80

Table 4-5: IBM System x3950 M2 Data Distribution Benchmark Configuration

4.5. 60-Day Space Calculations

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

60-Day Space Computation

All data sizes in MB unless otherwise stated

Warehouses	96,000
Measured TpmC	1,200,632

Table	Rows	Table	Index	5% Space	Total Space
Warehouse	96,000	60	0	3	63
District	960,000	150	0	8	158
Item	100,000	10	0	1	11
Stock	9,600,000,000	3,125,150	0	156,258	3,281,408
Customer	2,880,000,000	2,250,120	138,580	119,435	2,508,135
New-Order	864,000,000	66,360	0	3,318	69,678
Orders	2,880,000,000	110,321	80,500	0	190,821
Order-Line	28,800,000,000	1,929,484	0	0	1,929,484
History	2,880,000,000	177,360	0	0	177,360
Additional Overhead		1,660,295			1,660,295

Free Space	1,064,098		
Dynamic Space	2,217,165		
Static Space	7,600,247		
Daily Growth	443,667		
Daily Spread	398,599		
		<u>30 Minute log Computations</u>	
		Log Written (KB)	80,682,470
		New-Order Txns	36,018,960
		Log Written per New-Order (KB)	2.25

Data Storage Requirement

60 Days (MB)	58,136,152
60 Days (GB)	56,774

Log Storage Requirement

8 Hours (GB)	1,236.61
--------------	----------

Disk Sizing

Disk Type	Formatted		SUT		Priced	
	Capacity (GB)	# of Disks	Capacity (GB)	# of Disks	Capacity (GB)	# of Disks
DB RAID0 73.4GB	73.40	1,920	140,928	1,920	140,928	1,920
LOG RAID10	300.00	10	1,500	10	1,500	10
OS SAS 73GB	73.40	1	73	1	73	1

Total Capacity	142,501
-----------------------	---------

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 %	0.920	0.900	0.920	0.30/0.31	1.020	0.282
Average	0.409	0.389	0.404	0.154/0.21	0.484	.153
Maximum	9.297	9.282	9.281	8.984/2.41	9.407	9.468
Think Times						
Minimum	0	0	0	0	0	N/A
Average	12.036	12.036	10.036	5.038	5.038	N/A
Maximum	120.313	120.313	100.313	50.313	50.313	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	18.016	3.016	2.016	2.016	2.016	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.

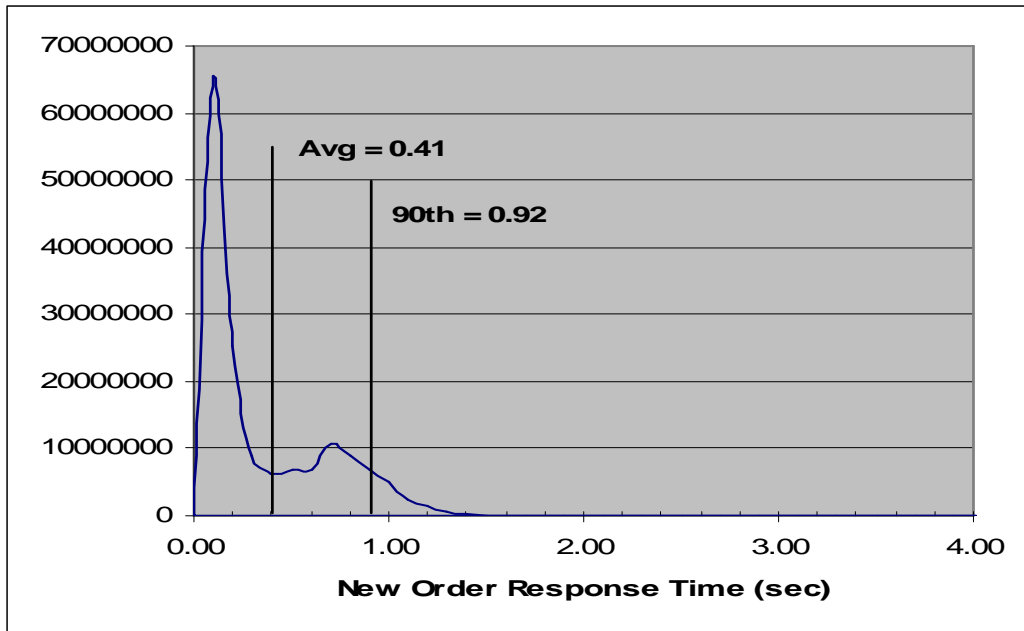


Figure 5-1: New-Order Response Time Distribution

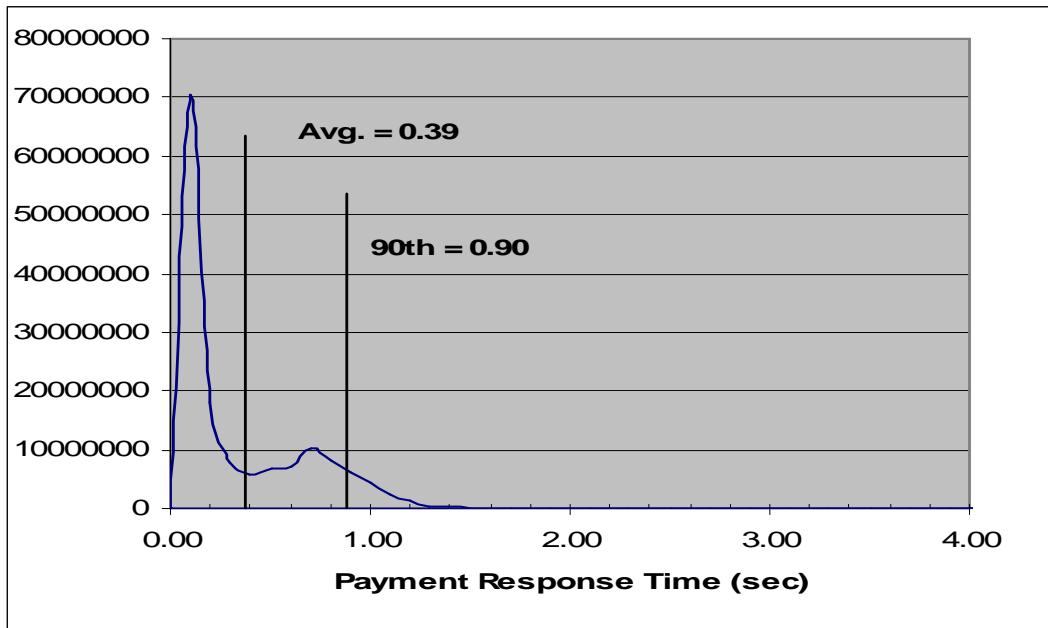


Figure 5-2: Payment Response Time Distribution

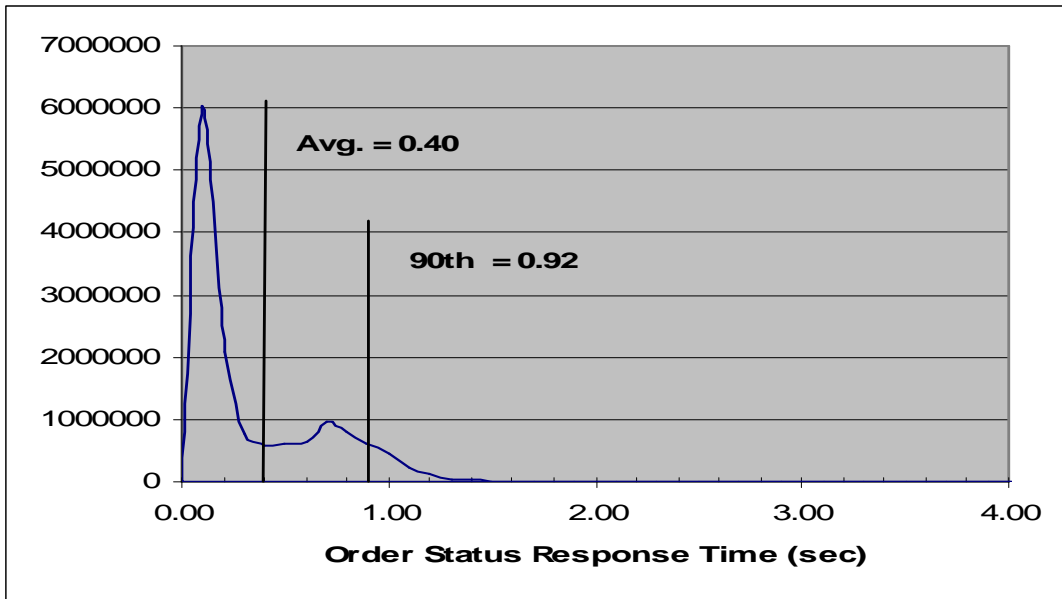


Figure 5-3: Order-Status Response Time Distribution

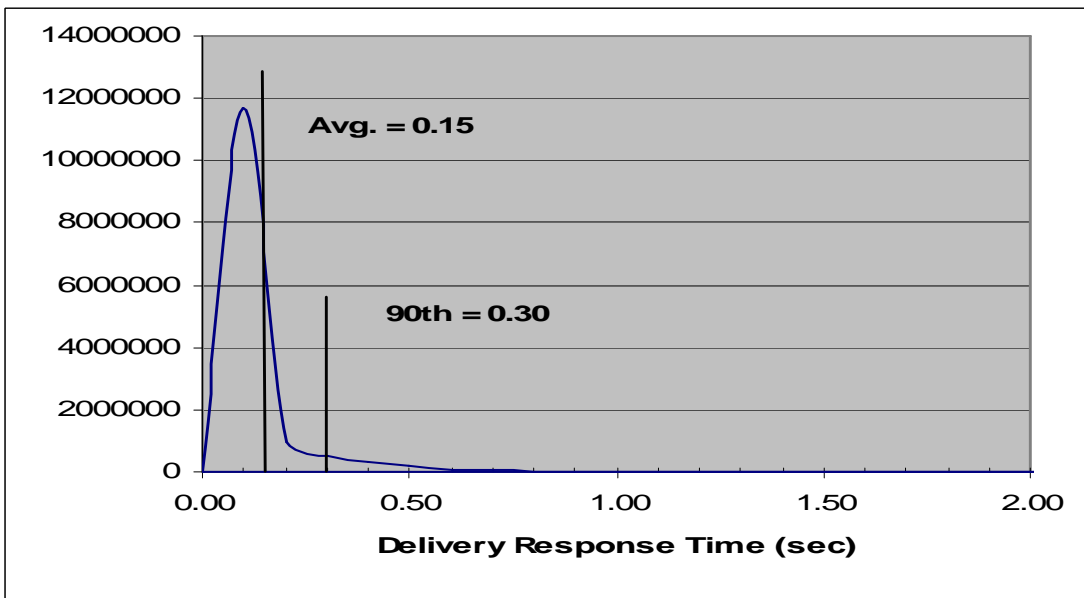


Figure 5-4: Delivery (Interactive) Response Time Distribution

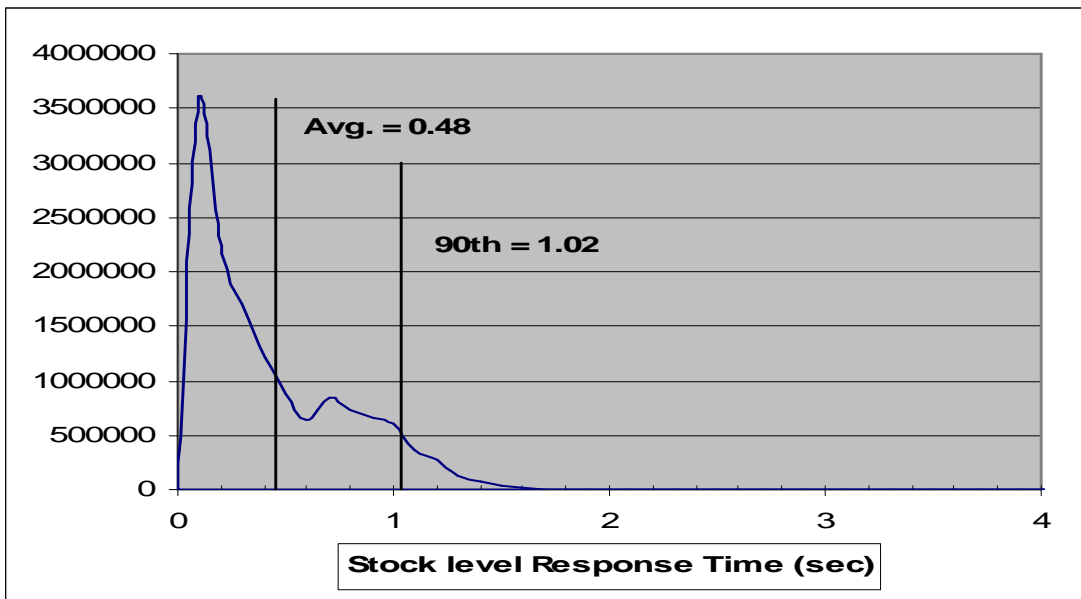


Figure 5-5: Stock Level Response Time Distribution

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.

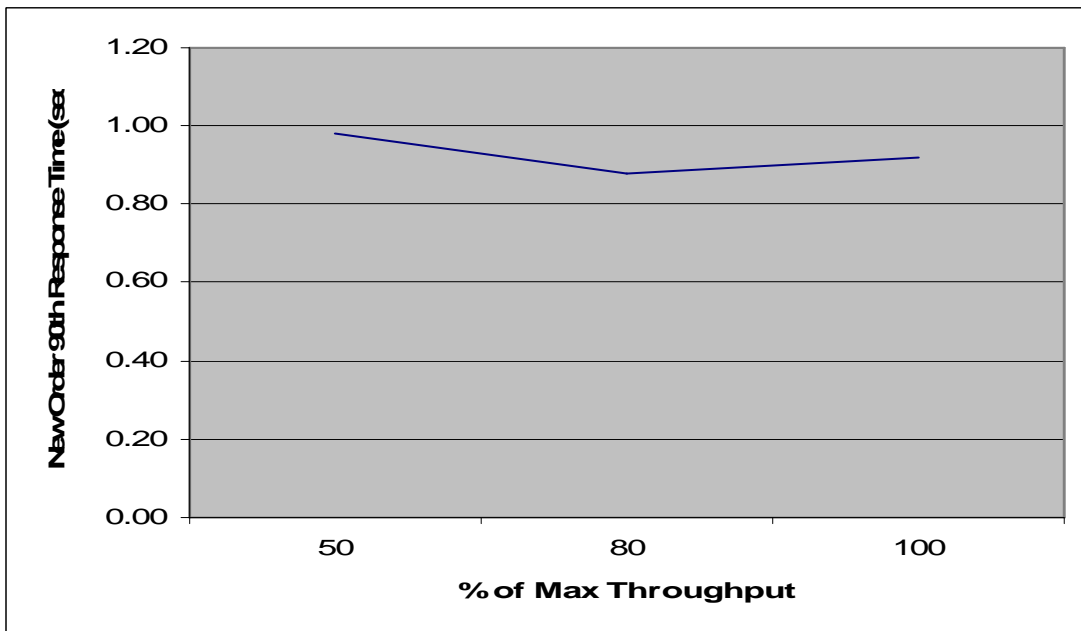


Figure 5-6: New-Order Response Time vs. Throughput

5.5. Think Time Frequency Distribution

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

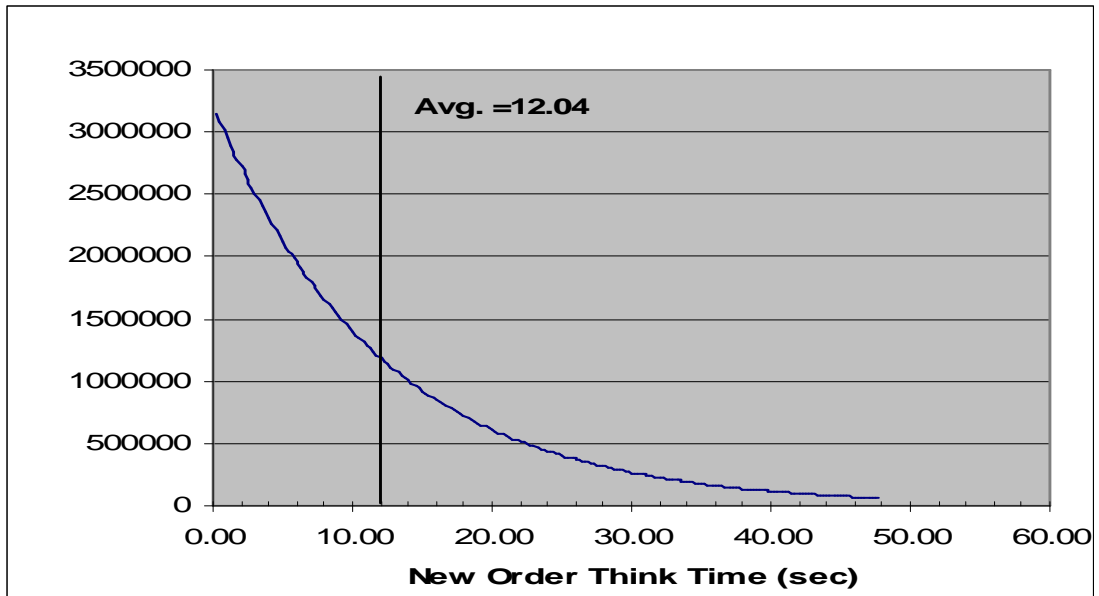


Figure 5-7: New-Order Think Time Distribution

5.6. Throughput versus Elapsed Time

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

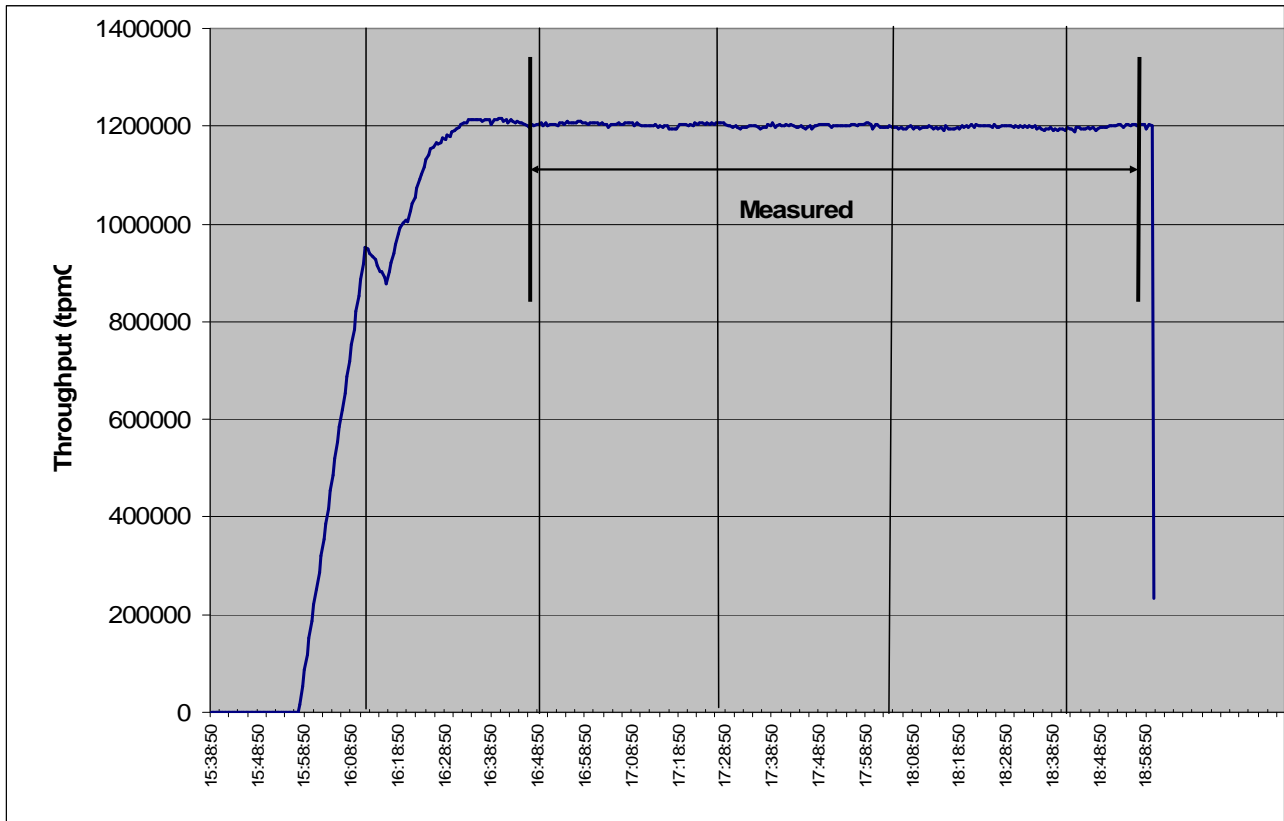


Figure 5-8: New-Order Throughput vs. Elapsed Time

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 and 10 minute 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 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](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 and 10 minute 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. 96,000 warehouses were configured. A rampup time of 18 minutes was specified, along with a run time of two hours and 10 minutes.

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 US list prices.

Discounts are based on US list prices and for similar quantities and configurations. A discount of 23.36% 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 December 10, 2008.

Description	Part Number	Order Date/ Availability	Order Method	Price Verification
IBM System x3950 M2 with Intel Xeon Processor X7460	7233-6SU	9-16-08/ 10-10-08	See Note 1	See Note 1
Intel Xeon Processor X7460	44E4473	9-16-08/ 10-10-08	See Note 1	See Note 1
DB2 ESE 9.5 FP3		8-19-08/ 12-10-08	See Note 2	See Note 2

Note 1: This component is not immediately orderable. For price verification before order date, call 1-888-746-7426, ext. 5821.

Note 2: DB2 ESE 9.5 is orderable and available as of submission. Upgrade to FP3 when available on 12/10 is included.

7.4. Statement of tpmC and Price/Performance

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

System	tpmC	3-year System Cost	\$/tpmC	Availability Date
IBM System x3950 M2	1,200,632	\$2,386,768 USD	\$1.99 USD	December 10, 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:



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

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

August 8, 2008

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

Platform: IBM System x3950 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 x3950 M2				
8 x Intel Xeon X7460 (2.66GHz)	512 GB (3 x 3MB L2)	1921 x 73 GB 15K rpm SAS 10 x 300 GB 15K rpm SAS	0.92 Seconds	1,200,632.7
24 Clients: IBM System x3250 M2 (each with)				
1 x Intel Xeon X3350 QC (2.66 GHz)	1 GB (2 x 6 MB L2)	1 x 73.4 GB SAS	n/a	n/a
8 Clients: IBM System x3550 (each with)				
1 x Intel Xeon X5130 DC (2.0 GHz)	1 GB (2 x 4 MB L2)	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.

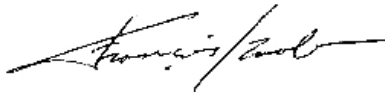
The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- 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 130 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:

Client Substitution - The tested configuration included (24) IBM System x3250, configured with a 2.13GHz Intel x3210. These clients were substituted one for one in the priced configuration by the same model IBM System x3250 upgraded to a 2.66GHz Intel x3250. Based on data analysis done for each type of client, it is my opinion that this substitution has no significant effect on performance.

Respectfully Yours,



François Raab, President

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
LD_FLAGS_EXEC=
LD_FLAGS_SHLIB=DLL
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB) /DEF:rtpcc.def
LD_FLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib /LIBPATH:"C:\Program
Files\Microsoft Visual Studio\VC98\Lib" db2api.lib winmm.lib
LD_FLAGS_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);
#endif

    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_sta
te
        , :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_stat
e
        , :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 ;
}
#endif 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 ;
#endif DEBUGIT
ord_debug(ordstat, in_ordstat, "Client before SP call");
#endif /* DEBUGIT */
#endif 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 );
#endif 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 )
{
    // Propagate 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 ;
}
#endif 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$(OBJEXT) tpcctdbg$(OBJEXT)
UTIL_OBJ_DB2 = tpcctcx$(OBJEXT)

#
#####
#####
# User Targets
#
#####
all: dbgen connect $(UTIL_OBJ_DB2) disconnect

dbgen: $(UTIL_OBJ)

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

#
#####
#####
# Helper Targets
#
#####
connect:
  - db2 connect to $(TPCC_DBNAME)

disconnect:
  - db2 connect reset
  - db2 terminate

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

#
#####
#####
# Build Rules
#
#####
.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
  @echo "Prepping $.sqc"
  -db2 prep $.sqc $(PRP_OPTS)

```

```
@echo "Binding $.bnd"
db2 bind $.bnd $(BND_OPTS)
```

```
#
#####
#####
# Dependencies
#
#####
#####
```

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

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

Src.Common/tpccctx.sqc

```
/*
** Licensed Materials - Property of IBM
**
```

```
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
```

```
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2005
** All Rights Reserved.
**
```

```
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****/
```

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

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

```
int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char *in_dbname, char *in_username, char
    *in_password);
int disconnect_from_TM(void);
int create_context();
int destroy_context();
int attach_context(void*);
int detach_context(void*);
int get_context(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;
```

```
SQLCODE = create_context();
if (SQLCODE != 0) { return SQLCODE; }
```

```
/* 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);
```

```
SQLCODE = destroy_context();
if (SQLCODE != 0) { return SQLCODE; }
```

```
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);
}
```

```
SQLCODE = destroy_context();
if (SQLCODE != 0) { return SQLCODE; }
```

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

```
int create_context(void)
```

```
{
    SQL_STRUCTURE sqlca sqlca;
    void *ctx;

    sqlcSetTypeCtx(SQL_CTX_MULTI_MANUAL);
    sqlcBeginCtx(&ctx, SQL_CTX_BEGIN_ALL, NULL, &sqlca);
```

```
if (SQLCODE != 0) {
    sqlerror( CLIENT_SQL, "CREATE", __FILE__, __LINE__, &sqlca);
    return SQLCODE;
}
```

```
return 0;
}
```

```
int attach_context(void *ctx)
{
```

```
SQL_STRUCTURE sqlca sqlca;
```

```
sqlcAttachToCtx(ctx, NULL, &sqlca);
```

```
if (SQLCODE != 0) {
    sqlerror( CLIENT_SQL, "ATTACH", __FILE__, __LINE__, &sqlca);
    return SQLCODE;
}
```

```
return 0;
}
```

```
int detach_context(void *ctx)
```

```
{
    SQL_STRUCTURE sqlca sqlca;

    sqlcDetachFromCtx(ctx, NULL, &sqlca);
```

```
if (SQLCODE != 0) {
    sqlerror( CLIENT_SQL, "DETACH", __FILE__, __LINE__, &sqlca);
    return SQLCODE;
}
```

```
return 0;
}
```

```
int destroy_context(void)
```

```
{
    SQL_STRUCTURE sqlca sqlca;
    void *ctx;
```

```
SQLCODE = get_context(&ctx);
if (SQLCODE) { return SQLCODE; }
```

```
sqlcEndCtx(&ctx, SQL_CTX_END_ALL, NULL, &sqlca);
```

```
if (SQLCODE != 0) {
    sqlerror( CLIENT_SQL, "DESTROY", __FILE__, __LINE__, &sqlca);
    return SQLCODE;
}
```

```
return 0;
}
```

```
int get_context(void **ctx)
```

```
{
    SQL_STRUCTURE sqlca sqlca;
```

```
sqlcGetCurrentCtx(ctx, NULL, &sqlca);
```

```
if (SQLCODE != 0) {
    sqlerror( CLIENT_SQL, "GETCTX", __FILE__, __LINE__, &sqlca);
    return SQLCODE;
}
```

```
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.
*****
/*
 * tcdbg.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_tmstamp(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_tmstamp(&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, "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, "\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_tmstamp(&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, "\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, "\tts_O_ID[%d] = %d\n",
        j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\t)\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_tmstamp(&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, "\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, "\tts_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, "\tts_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, "\tts_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");

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 = %0.2f\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, "\tts_I_NAME[%d] = %s\n",
        j, neword_ptr->item[j].s_I_NAME);
    fprintf(debug_fp, "\tts_I_PRICE[%d] = %0.2f\n",
        j, neword_ptr->item[j].s_I_PRICE);
    fprintf(debug_fp, "\tts_OL_AMOUNT[%d] = %0.2f\n",
        j, neword_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "\tts_S_QUANTITY[%d] = %d (%X)\n",
        j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr->item[j].s_S_QUANTITY);
    fprintf(debug_fp, "\tts_brand_generic[%d] = %c\n",
        j, neword_ptr->item[j].s_brand_generic);
}
}
fprintf(debug_fp, "\t)\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_tmstamp(&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, "\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");

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, "\n");
items = ordstat_ptr->s_ol_cnt;
for (j = 0; j < items; j++) {
    if (j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "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, "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, "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, "ts_OL_AMOUNT[%d] = %.2f\n",
             j, ordstat_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "ts_OL_DELIVERY_D[%d] = %s\n",
             j, ordstat_ptr->item[j].s_OL_DELIVERY_D_time);
}
fprintf(debug_fp, "\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_tmstamp(&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, "\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");

    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");
    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_tmstamp(&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, "\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_tmstamp(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>
typedef __int16 int16_t;
typedef __int32 int32_t;
typedef __int64 int64_t;

#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.
*/
*****

#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[20];
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);

extern int create_context(void);
extern int destroy_context(void);
extern int get_context(void**);
extern int attach_context(void*);
extern int detach_context(void*);

#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_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
 *
 */

#ifdef __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 DARVERSION 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 authorization 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\sqlib
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. 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 - 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_tmstamp(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_tmstamp(&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 ORDDSTAT_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, "sqlerr: ");
for(j = 0; j < 8; j++)
fprintf(err_fp, "%c", psqlca->sqlerr[j]);

```

```

fprintf(err_fp, "\n");

fprintf(err_fp, "sqlerr: ");
for(j = 0; j < 8; j++)
fprintf(err_fp, "%d", psqlca->sqlerr[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 *in_delivery_ptr,
struct in_delivery_struct *in_delivery,
char *msg)
{
char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

InitializeDebug();
strcpy(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 *in_delivery_ptr,
struct in_delivery_struct *in_delivery,
char *filename,
char *msg)
{
FILE *debug_fp;
char timeStamp[27];
int j;

current_tmstamp(&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, "\tts_O_ID[%d] = %d\n",
j, delivery_ptr->s_O_ID[j]);
}
fprintf(debug_fp, "\t}\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();
strcpy(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_tmstamp(&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_newword->s_transtatus, in_newword->s_transtatus);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
// in_newword->duplicate_items, in_newword->duplicate_items);

fprintf(debug_fp, "titems {\n");
items = in_newword->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "tts_OL_I_ID[%d] = %d (%X)\n",
        j, in_newword->in_item[j].s_OL_I_ID, in_newword->in_item[j].s_OL_I_ID);
    fprintf(debug_fp, "tts_OL_SUPPLY_W_ID[%d] = %d (%X)\n",
        j, in_newword->in_item[j].s_OL_SUPPLY_W_ID, in_newword-
>in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp, "tts_OL_QUANTITY[%d] = %d (%X)\n",
        j, in_newword->in_item[j].s_OL_QUANTITY, in_newword-
>in_item[j].s_OL_QUANTITY);
}
fprintf(debug_fp, "tj}\n\n");

fprintf(debug_fp, "out_newword_struct {\n");
fprintf(debug_fp, "tts_C_LAST = %s\n",
        newword_ptr->s_C_LAST);
fprintf(debug_fp, "tts_C_CREDIT = %s\n",
        newword_ptr->s_C_CREDIT);
fprintf(debug_fp, "tts_W_TAX = %04.4f\n",
        newword_ptr->s_W_TAX);
fprintf(debug_fp, "tts_D_TAX = %04.4f\n",
        newword_ptr->s_D_TAX);
fprintf(debug_fp, "tts_C_DISCOUNT = %04.4f\n",
        newword_ptr->s_C_DISCOUNT);
fprintf(debug_fp, "tts_O_ID = %d (%X)\n",
        newword_ptr->s_O_ID, newword_ptr->s_O_ID);
fprintf(debug_fp, "tts_O_OL_CNT = %d (%X)\n",
        newword_ptr->s_O_OL_CNT, newword_ptr->s_O_OL_CNT);
fprintf(debug_fp, "tts_O_ENTRY_D = %s\n",
        newword_ptr->s_O_ENTRY_D_time);
fprintf(debug_fp, "tts_total_amount = %2f\n",
        newword_ptr->s_total_amount);
fprintf(debug_fp, "tts_transtatus = %d (%X)\n",
        newword_ptr->s_transtatus, newword_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
        newword_ptr->deadlocks, newword_ptr->deadlocks);

// fprintf(debug_fp, "tts_W_ID = %d (%X)\n",
// newword_ptr->s_W_ID, newword_ptr->s_W_ID);
// fprintf(debug_fp, "tts_D_ID = %d (%X)\n",
// newword_ptr->s_D_ID, newword_ptr->s_D_ID);
// fprintf(debug_fp, "tts_all_local = %d (%X)\n",
// newword_ptr->s_all_local, newword_ptr->s_all_local);
// fprintf(debug_fp, "tduplicate_items= %d (%X)\n",
// newword_ptr->duplicate_items, newword_ptr->duplicate_items);

fprintf(debug_fp, "titems {\n");
items = newword_ptr->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp, "\n");
    fprintf(debug_fp, "tts_I_NAME[%d] = %s\n",
        j, newword_ptr->item[j].s_I_NAME);
    fprintf(debug_fp, "tts_I_PRICE[%d] = %2f\n",
        j, newword_ptr->item[j].s_I_PRICE);
    fprintf(debug_fp, "tts_OL_AMOUNT[%d] = %2f\n",
        j, newword_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "tts_S_QUANTITY[%d] = %d (%X)\n",
        j, newword_ptr->item[j].s_S_QUANTITY, newword_ptr-
>item[j].s_S_QUANTITY);
    fprintf(debug_fp, "tts_brand_generic[%d] = %c\n",
        j, newword_ptr->item[j].s_brand_generic);
}
}
```

```
fprintf(debug_fp, "tj}\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_tmstamp(&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, "tts_W_ID = %d (%X)\n",
        in_ordstat->s_W_ID, in_ordstat->s_W_ID);
    fprintf(debug_fp, "tts_D_ID = %d (%X)\n",
        in_ordstat->s_D_ID, in_ordstat->s_D_ID);
    fprintf(debug_fp, "tts_C_ID = %d (%X)\n",
        in_ordstat->s_C_ID, in_ordstat->s_C_ID);
    fprintf(debug_fp, "tts_C_LAST = %s\n",
        in_ordstat->s_C_LAST);
    fprintf(debug_fp, "tj}\n\n");

    fprintf(debug_fp, "out_ordstat_struct {\n");
    fprintf(debug_fp, "tts_C_ID = %d (%X)\n",
        ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);
    fprintf(debug_fp, "tts_C_FIRST = %s\n",
        ordstat_ptr->s_C_FIRST);
    fprintf(debug_fp, "tts_C_MIDDLE = %s\n",
        ordstat_ptr->s_C_MIDDLE);
    fprintf(debug_fp, "tts_C_LAST = %s\n",
        ordstat_ptr->s_C_LAST);
    fprintf(debug_fp, "tts_C_BALANCE = %2f\n",
}
```

```
ordstat_ptr->s_C_BALANCE);
fprintf(debug_fp, "tts_O_ID = %d (%X)\n",
        ordstat_ptr->s_O_ID, ordstat_ptr->s_O_ID);
fprintf(debug_fp, "tts_O_ENTRY_D = %s\n",
        ordstat_ptr->s_O_ENTRY_D_time);
fprintf(debug_fp, "tts_O_CARRIER_ID = %d (%X)\n",
        ordstat_ptr->s_O_CARRIER_ID, ordstat_ptr->s_O_CARRIER_ID);
fprintf(debug_fp, "tts_ol_cnt = %d (%X)\n",
        ordstat_ptr->s_ol_cnt, ordstat_ptr->s_ol_cnt);
fprintf(debug_fp, "tts_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, "tts_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, "tts_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, "tts_OL_QUANTITY[%d] = %d (%X)\n",
        j, ordstat_ptr->item[j].s_OL_QUANTITY, ordstat_ptr-
>item[j].s_OL_QUANTITY);
    fprintf(debug_fp, "tts_OL_AMOUNT[%d] = %2f\n",
        j, ordstat_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "tts_OL_DELIVERY_D[%d] = %s\n",
        j, ordstat_ptr->item[j].s_OL_DELIVERY_D_time);
}
fprintf(debug_fp, "tj}\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_tmstamp(&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_tmstamp(&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\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_tmstamp(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

LDLFLAGS = $(LDLFLAGS_STORP) $(LDLFLAGS_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 -td% -vf uncat-proc.ddl +o -z uncat-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 -td% -vf uncat-proc.ddl +o -z uncat-proc.out

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

unexplain:
  - db2 -td% -vf $(TPCC_ROOT)$(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; db2explan -d $(TPCC_DBNAME) -c
  $(TPCC_SCHEMA) -p TPCC_ALL -s 0 -g -o TPCC_ALL.explan.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 .sql

# 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 $*.sql"
-db2 prep $*.sql $(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) $(LDLFLAGS) $(UTIL_OBJ) tpcc_all_sql.o
  $(LDLFLAGS_OUT)$(

#
#####
#####
# 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 retrieve 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
) 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
) 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 retrieve 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

```



```

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 inputItemCnt;
    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_newword->s_W_ID
#define d_id in_newword->s_D_ID
#define c_id in_newword->s_C_ID

#define inputItemCount in_newword->s_O_OL_CNT

#define allLocal in_newword->s_all_local

// Redirected output fields

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

// This output field becomes an input field to order_line

#define next_o_id newword->s_O_ID

// item price/name

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

#define ol_quantity0 in_newword->in_item[ 0 ].s_OL_QUANTITY
#define ol_quantity1 in_newword->in_item[ 1 ].s_OL_QUANTITY
#define ol_quantity2 in_newword->in_item[ 2 ].s_OL_QUANTITY
#define ol_quantity3 in_newword->in_item[ 3 ].s_OL_QUANTITY

```

```

#define ol_quantity4 in_newword->in_item[ 4 ].s_OL_QUANTITY
#define ol_quantity5 in_newword->in_item[ 5 ].s_OL_QUANTITY
#define ol_quantity6 in_newword->in_item[ 6 ].s_OL_QUANTITY
#define ol_quantity7 in_newword->in_item[ 7 ].s_OL_QUANTITY
#define ol_quantity8 in_newword->in_item[ 8 ].s_OL_QUANTITY
#define ol_quantity9 in_newword->in_item[ 9 ].s_OL_QUANTITY
#define ol_quantity10 in_newword->in_item[ 10 ].s_OL_QUANTITY
#define ol_quantity11 in_newword->in_item[ 11 ].s_OL_QUANTITY
#define ol_quantity12 in_newword->in_item[ 12 ].s_OL_QUANTITY
#define ol_quantity13 in_newword->in_item[ 13 ].s_OL_QUANTITY
#define ol_quantity14 in_newword->in_item[ 14 ].s_OL_QUANTITY

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

FROM ( VALUES
( SMALLINT( 1 ) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
, I_SUPPLY_W_ID )
) AS X ( OL_NUMBER , I_ID , I_QTY
) 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 )
, I_SUPPLY_W_ID )
) AS X ( OL_NUMBER , I_ID , I_QTY
) 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
)

```

```

        ) 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 )

) 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
;

```

```

), 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 )

) 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
;

```

```

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_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
)

```

```

) 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
      ,OL_I_ID
      ,OL_SUPPLY_W_ID
      ,OL_DELIVERY_D
      ,OL_QUANTITY
      ,OL_AMOUNT
      ,OL_DIST_INFO
      ,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
) 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

```

```

        ) 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

```

```

, 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_newword = (struct in_newword_struct *) pin ;
newword = (struct out_newword_struct *) pout ;

#ifdef DEBUGIT
new_debug( newword, in_newword, "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_newword-
>s_O_OL_CNT ; inputItemArrayIndex++)
{
i_priceArray[ inputItemArrayIndex ] = 0 ;
}

newword->deadlocks = -1 ;

retry_tran:
newword->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( NEWWORD_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 )
{
newword->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
        ,:inputItemCnt
        ,: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, "inputItemCnt=%d, :next_o_id=%d, :d_id=%d, :w_id=%d",
    inputItemCnt, 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, /*0-9*/
    8,8,8,8,8,8,8,8, /*10-19*/
    8,8,8,8,8,8,8,8, /*20-29*/
    8,8,8,8,8,8,8,8, /*30-39*/
    8,8,8,8,8,8,8,8, /*40-49*/
    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, /*100-109*/
    8,8,8,8,8,8,8,8,8, /*110-119*/
    8,8,8,8,8,8,8,8,8, /*120-129*/
    8,8,8,8,8,8,8,8,8, /*130-139*/
    8,8,8,8,8,8,8,8,8, /*140-149*/
    8,8,8,8,8,8,8,8,8, /*150-159*/
    8,8,8,8,8,8,8,8,8, /*160-169*/
    8,8,8,8,8,8,8,8,8, /*170-179*/
    8,8,8,8,8,8,8,8,8, /*180-189*/
    8,8,8,8,8,8,8,8,8, /*190-199*/
    8,8,8,8,8,8,8,8,8, /*200-209*/
    8,8,8,8,8,8,8,8,8, /*210-219*/
    8,8,8,8,8,8,8,8,8, /*220-229*/
    8,8,8,8,8,8,8,8,8, /*230-239*/
    8,8,8,8,8,8,8,8,8, /*240-249*/
    8,8,8,8,8,8,8,8,8, /*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_SQLEERROR ;

EXEC SQL ROLLBACK WORK ;

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

goto mexit;
}

// -----
// Delivery SERVER
// -----

#undef d_id
#undef c_id
#undef w_id
#undef o_carrier_id
#undef ol_delivery_d

SQL_API_RC delivery_internal ( char * pin, char * pout )
{
struct in_delivery_struct * in_delivery = (struct in_delivery_struct *) pin ;
struct out_delivery_struct * delivery = (struct out_delivery_struct *) pout ;

struct sqlca sqlca ;

int storedProcRc ;

```



```

short district_id ;
sqlint32 customer_id ;

EXEC SQL BEGIN DECLARE SECTION;

// input

###sqlint32 w_id ;
###short d_id ;
###sqlint32 c_id ;
###short o_carrier_id ;
###sqlint64 ol_delivery_d ;

// output

short no_o_id_indicator = 0 ;
sqlint32 no_o_id ;

EXEC SQL END DECLARE SECTION;

#define d_id district_id
#define c_id customer_id

#define w_id in_delivery->s_W_ID
#define o_carrier_id in_delivery->s_O_CARRIER_ID
#define ol_delivery_d in_delivery->s_O_DELIVERY_D_time

delivery->deadlocks = -1 ;

#ifdef DEBUGIT
del_debug( delivery, in_delivery, "SP upon entry");
#endif

// Deadlock Handling
// -----
// Since we COMMIT inside the for() loop, we must take special
// care while handling deadlocks. This is best explained by
// an example.
//
// Assume we deadlock on d_id=6. This means that an order from the
// first 5 districts have already been delivered. We will then
// restart the loop (retry_tran). However, the loop will restart
// at d_id = 1! This means that the second (and all subsequent)
// time through the loop, we will deliver orders for districts that
// have already been delivered, with the net result being more than
// 10 orders being delivered.
//
// The solution to this problem is to initialize the starting point
// of the loop "before" the retry_tran label. This will ensure that
// if we deadlock, we will restart the loop with the same district
// that we deadlocked on, and we won't deliver any extra orders.
//
// NOTE: If we ever change this back to one COMMIT per transaction
// (instead of one COMMIT per iteration), then the initialization
// of d_id must be moved back into the for loop. (A rollback due
// to deadlock in this case would rollback all delivered orders so
// far, so we'd need to re-deliver them all on the next iteration.)

d_id = 1;

retry_tran:

delivery->deadlocks++;

for ( ; d_id <= DISTRICTS_PER_WAREHOUSE ; d_id++)
{
no_o_id = 0 ;
no_o_id_indicator = 0 ;

EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

```

```

SELECT O_ID

INTO :no_o_id :no_o_id_indicator

FROM TABLE ( DEL( :w_id , :d_id , :o_carrier_id ) ) AS T ;

COMMIT ;

END COMPOUND ;

if ( sqlca.sqlcode == 0 )
{
delivery->s_O_ID[ d_id - 1 ] = no_o_id ;
}
else
{
DLCHK( retry_tran );

sqlerror( DELIVERY_SQL , "DELIVERY", __FILE__, __LINE__, &sqlca);
goto ferror ;
}

delivery->s_transtatus = TRAN_OK ;

mexit:

if ( sqlca.sqlcode >= 0 )
{
storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
storedProcRc = SQLZ_DISCONNECT_PROC ;
}

#ifdef DEBUGIT
del_debug( delivery, in_delivery, "SP prior to return");
#endif

return ( storedProcRc ) ;

ferror:

delivery->s_transtatus = FATAL_SQLERROR ;

EXEC SQL ROLLBACK WORK ;

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

goto mexit ;
}

// -----
// Stored Procedure Stubs
// -----

SQL_API_RC SQL_API_FN news( char *pin, char *pout )
{
return new_order_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN ords( char *pin, char *pout )
{

```

```

return order_status_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN dels ( char * pin, char * pout )
{
return delivery_internal( pin, pout ) ;
}

```

Src.Srv/uncat-func.ddl

```

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

-- uncat-func.ddl - Drop table function DDL
--
--
-- STOCK LEVEL
DROP SPECIFIC FUNCTION STOCK_LEVEL %
-- DELIVERY
DROP SPECIFIC FUNCTION DELIVERY %
-- ORDER STATUS
DROP SPECIFIC FUNCTION ORD_C_LAST %
DROP SPECIFIC FUNCTION ORD_C_ID %
-- PAYMENT
DROP SPECIFIC FUNCTION PAY_C_LAST %
DROP SPECIFIC FUNCTION PAY_C_ID %
-- NEW ORDER
DROP SPECIFIC FUNCTION NEW_OL_ALL %
DROP SPECIFIC FUNCTION NEW_OL_LOCAL %
DROP SPECIFIC FUNCTION NEW_WH %

DROP PROCEDURE news
      (varchar(262),varchar(682));
DROP PROCEDURE news
      (varchar(270),varchar(662));
DROP PROCEDURE news;

```

Src.Srv/uncat-proc.ddl

```

DROP PROCEDURE pays;

DROP PROCEDURE ords
      (varchar(42),varchar(822));
DROP PROCEDURE ords
      (varchar(42),varchar(446));
DROP PROCEDURE ords;

DROP PROCEDURE dels
      (varchar(14),varchar(50));
DROP PROCEDURE dels
      (varchar(22),varchar(50));
DROP PROCEDURE dels;

DROP PROCEDURE stks;

```

include/db2tpcc.h

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

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

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

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

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

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

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

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*
** NURand Constants
** C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.
*/

#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[20];
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 */

#ifdef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 96000
#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 l=0x12345678; SWAP_BYTE(l); l => 0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Addr + 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 authorization 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/aff_dunn2cpu_m4.cfg
DB2_SELUDL_COMM_BUFFER=Y
DB2_USE_ALTERNATE_PAGE_CLEANSING=YES
DB2_MAX_NON_TABLE_LOCKS=500
DB2_RCT_FEATURES=GROUPUPDATE=ON
DB2_TRUSTED_BINDIN=ON
DB2_KEEPTABLELOCK=CONNECTION
DB2_NO_FORK_CHECK=ON
DB2_ALLOCATION_SIZE=8388608
DB2_FMP_COMM_HEAPSZ=0
DB2_APM_PERFORMANCE=ALL
DB2_PINNED_BP=ON
DB2_SELECTIVITY=ON
DB2ASSUMEUPDATE=ON
DB2CHECKCLIENTINTERVAL=0
DB2_HASH_JOIN=OFF
DB2CHKSQDLA=OFF
DB2MEMDISCLAIM=NO
DB2_COLLECT_TS_REC_INFO=false
DB2COMM=tcip
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) = 129789024
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) = 6000
Utilities heap size (4KB) (UTIL_HEAP_SZ) = 5000
Buffer pool size (pages) (BUFFPAGE) = 1000
SQL statement heap (4KB) (STMTHEAP) = AUTOMATIC
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) = 2000
Average number of active applications (AVG_APPLS) = AUTOMATIC
Max DB files open per application (MAXFILOP) = 2200
```

```
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 = S0000004.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) = 4287
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 = Database Server with local clients
```

```
Database manager configuration release level = 0x0c00
```

```
CPU speed (millisec/instruction) (CPUSPEED) = 1.889377e-07
```

```
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/sql/lib/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 List 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) = YES
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) = 131932709
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 (SVCSNAME) =
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

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) = AUTOMATIC
No. of int. communication channels (FCM_NUM_CHANNELS) = AUTOMATIC
db2start/db2stop timeout (min) (START_STOP_TIME) = 10

```

aff2_dunn2cpu_m4.cfg

```

<!-- This policy is valid -->
<RESOURCE_POLICY>
<DATABASE_RESOURCE_POLICY>
  <DBNAME>=tpcc</DBNAME>
  <METHOD>=CPUMASK</METHOD>

<RESOURCE_BINDING>
<RESOURCE>0x000000111111</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50021</SERVICE_NAME>
<SERVICE_NAME>50022</SERVICE_NAME>
<BUFFERPOOL_BINDING>
  <NUM_CLEANERS>4</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_ID>61</BUFFERPOOL_ID>
  <BUFFERPOOL_ID>69</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>0x000000222222</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50023</SERVICE_NAME>
<SERVICE_NAME>50024</SERVICE_NAME>
<BUFFERPOOL_BINDING>
  <NUM_CLEANERS>4</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_ID>62</BUFFERPOOL_ID>
  <BUFFERPOOL_ID>70</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>0x000000444444</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50025</SERVICE_NAME>
<SERVICE_NAME>50026</SERVICE_NAME>
<BUFFERPOOL_BINDING>
  <NUM_CLEANERS>4</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_ID>63</BUFFERPOOL_ID>
<BUFFERPOOL_ID>71</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>0x000000888888</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50027</SERVICE_NAME>
<SERVICE_NAME>50028</SERVICE_NAME>
<BUFFERPOOL_BINDING>
  <NUM_CLEANERS>4</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_ID>64</BUFFERPOOL_ID>
  <BUFFERPOOL_ID>72</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>0x111111000000</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50029</SERVICE_NAME>
<SERVICE_NAME>50030</SERVICE_NAME>
<BUFFERPOOL_BINDING>
  <NUM_CLEANERS>4</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_ID>65</BUFFERPOOL_ID>
  <BUFFERPOOL_ID>73</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>0x222222000000</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50031</SERVICE_NAME>
<SERVICE_NAME>50032</SERVICE_NAME>
<BUFFERPOOL_BINDING>
  <NUM_CLEANERS>4</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_ID>66</BUFFERPOOL_ID>
  <BUFFERPOOL_ID>74</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>0x444444000000</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50033</SERVICE_NAME>
<SERVICE_NAME>50034</SERVICE_NAME>
<BUFFERPOOL_BINDING>
  <NUM_CLEANERS>4</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_ID>67</BUFFERPOOL_ID>
<BUFFERPOOL_ID>75</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>0x888888000000</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>50035</SERVICE_NAME>
<SERVICE_NAME>50036</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>4</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_ID>68</BUFFERPOOL_ID>
<BUFFERPOOL_ID>76</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

```

tcipip parameters registry.reg

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="client1"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6
f,\
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"="client1"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableCMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"EnableTCPA"=dword:00000001
"EnableRSS"=dword:00000001
"EnableTCPC chimney"=dword:00000001
"MaxUserPort"=dword:0000ea50

[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-BC2690FD0FC}]
"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\DNS\RegisteredAdapters]

[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,00,31,00,31,0
0,\
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
"NetContextList"=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:4856e7c0
"T1"=dword:4856eec8
"T2"=dword:4856f40e
"LeaseTerminatesTime"=dword:4856f5d0
"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-BC2690FD0FC}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,30,00,2e,00,31,00,2e,00,31,00,2e,00,32,00,00,00,0
0,\
0
"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
"NetContextList"=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,00,30,00,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001

```

"DontAddDefaultGateway"=dword:00000000

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{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,
"DefaultGateway"=hex(7):00,00
"EnableDeadGW\Detect"=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,\
00,00,00,01,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00,00,\
00,00,00,00,00,06,00,00,00,00,00,00,00,00,01,00,00,00,06,00,00,02,00,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,02,00,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00
```

tpccCom.tpcc com settings.txt.txt

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"divvyLogPath"="c:\inetpub\wwwroot\tpcc\"
"divvyQueueLen"=dword:00009c40
"divvyThreads"=dword:0000000b
"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:00009c40
"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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
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 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client17

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client18

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client19

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client20

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client21

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client22

Transactions not supported
Enable Object pooling
Minimum pool size 44

Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client23

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client24

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client25

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client26

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client27

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client28

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client29

Transactions not supported

Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client30

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client31

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

tpcc software registry.reg client32

Transactions not supported
Enable Object pooling
Minimum pool size 44
Maximum pool size 44
Creation timeout 1,800,000,000
Enable Just in time activation
Concurrency Required

10.3. Linux Parameters

cmdline.txt

```
root=/dev/VG_RH/LogVol00 selinux=0 audit=0 console=ttyS0,115200  
elevator=noop splash=silent hugepages=253506
```

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/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 500 > /sys/class/scsi_host/host12/zio_timer
echo 500 > /sys/class/scsi_host/host13/zio_timer
echo 500 > /sys/class/scsi_host/host14/zio_timer
echo 500 > /sys/class/scsi_host/host15/zio_timer
echo 500 > /sys/class/scsi_host/host16/zio_timer
echo 500 > /sys/class/scsi_host/host17/zio_timer
echo 500 > /sys/class/scsi_host/host18/zio_timer
```

```
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
echo 1 > /sys/class/scsi_host/host12/zio
echo 1 > /sys/class/scsi_host/host13/zio
echo 1 > /sys/class/scsi_host/host14/zio
echo 1 > /sys/class/scsi_host/host15/zio
echo 1 > /sys/class/scsi_host/host16/zio
echo 1 > /sys/class/scsi_host/host17/zio
echo 1 > /sys/class/scsi_host/host18/zio
```

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 - qllogic 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
```

```
echo 00000000,00000000,00000000,00000010 > /proc/irq/122/smp_affinity
echo 00000000,00000000,00000000,00000020 > /proc/irq/138/smp_affinity
```

```
echo 00000000,00000000,00000000,00000040 > /proc/irq/154/smp_affinity
echo 00000000,00000000,00000000,00000080 > /proc/irq/170/smp_affinity
```

```
echo 00000000,00000000,00000000,00000100 > /proc/irq/186/smp_affinity
echo 00000000,00000000,00000000,00000200 > /proc/irq/202/smp_affinity
```

```
echo 00000000,00000000,00000000,00004000 > /proc/irq/218/smp_affinity
echo 00000000,00000000,00000000,00008000 > /proc/irq/234/smp_affinity
```

```
echo 00000000,00000000,00000000,01000000 > /proc/irq/59/smp_affinity
echo 00000000,00000000,00000000,02000000 > /proc/irq/75/smp_affinity
```

```
echo 00000000,00000000,00000004,00000000 > /proc/irq/107/smp_affinity
echo 00000000,00000000,00000008,00000000 > /proc/irq/91/smp_affinity
```

```
echo 00000000,00000000,00000010,00000000 > /proc/irq/123/smp_affinity
echo 00000000,00000000,00000020,00000000 > /proc/irq/139/smp_affinity
```

```
echo 00000000,00000000,00000040,00000000 > /proc/irq/155/smp_affinity
echo 00000000,00000000,00000080,00000000 > /proc/irq/171/smp_affinity
```

```
echo 00000000,00000000,00008000,00000000 > /proc/irq/203/smp_affinity
echo 00000000,00000000,00000000,00080000 > /proc/irq/187/smp_affinity
```

```
device=`cat /proc/interrupts | grep eth0 | awk -F : '{print $1}' | awk '{sub(/\/\ \t|+/, "");
print}'
print "eth0: echo 00000000,00000000,00000000,00100000 >
/proc/irq/$device/smp_affinity"
echo "00000000,00000000,00000000,00100000" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity
```

```
device=`cat /proc/interrupts | grep eth1 | awk -F : '{print $1}' | awk '{sub(/\/\ \t|+/, "");
print}'
print "eth1: echo 00000000,00000000,00000000,00200000 >
/proc/irq/$device/smp_affinity"
echo "00000000,00000000,00000000,00200000" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity
```

```
device=`cat /proc/interrupts | grep eth2 | awk -F : '{print $1}' | awk '{sub(/\/\ \t|+/, "");
print}'
print "eth2: echo 00000000,00002000,00000000,00000000 >
/proc/irq/$device/smp_affinity"
echo "00000000,00000000,00000200,00000000" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity
```

```
device=`cat /proc/interrupts | grep eth3 | awk -F : '{print $1}' | awk '{sub(/\/\ \t|+/, "");
print}'
print "eth3: echo 00000000,00004000,00000000,00000000 >
/proc/irq/$device/smp_affinity"
echo "00000000,00000000,00000400,00000000" > /proc/irq/$device/smp_affinity
cat /proc/irq/$device/smp_affinity
```

```
print "megasas: echo 00000000,00000000,00000000,00080000 >
/proc/irq/$device/smp_affinity"
echo "00000000,00000000,00000000,00080000" > /proc/irq/$device/smp_affinity
```

doit

```
#!/bin/sh
```

```
modprobe raw
```

```
/etc/rc.d/init.d/network stop
```

```
rmmod bnx2
rmmod e1000e
sleep 2
modprobe bnx2
ethtool -i eth0
ethtool -i eth1
ethtool -i eth2
ethtool -i eth3
sleep 2
modprobe e1000e
ethtool -i eth4
/etc/rc.d/init.d/network start
```

```
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
ethtool -C eth2 tx-usecs 160 tx-frames 0 rx-usecs 300 rx-frames 0
ethtool -C eth3 tx-usecs 160 tx-frames 0 rx-usecs 300 rx-frames 0
```

```
echo "affinitizing irq's"
/home/tpcc/tpc-c.ibm/affirq.sh
```

```
sysctl -p
/home/tpcc/tpc-c.ibm/setraw.sh
raw /dev/raw/raw1300 /dev/disk/by-id/scsi-
36000605b00036e570100517754876dcf7
sleep 10
chmod -R 777 /dev/raw/raw*
```

```
modprobe capability disable=1
modprobe capability enable=1
```

```
umount /sys/kernel/debug
umount /sys/kernel/security
```

/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 = 549755813888
kernel.shmall = 549755813888
fs.file-max = 524288
net.core.rmem_max = 131071
net.core.wmem_max = 131071
vm.hugetlb_shm_group = 102
```

setraw.sh

```
#!/bin/sh
modprobe raw
raw /dev/raw/raw1 /dev/disk/by-id/scsi-
3600a0b800034d63d00000f3b48519090-part1
raw /dev/raw/raw2 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-
part1
raw /dev/raw/raw3 /dev/disk/by-id/scsi-
3600a0b80003498e5000018474851a164-part1
raw /dev/raw/raw4 /dev/disk/by-id/scsi-
3600a0b800034d64c000011414851925c-part1
raw /dev/raw/raw5 /dev/disk/by-id/scsi-
3600a0b800034d6290000113a4852394a-part1
```

raw /dev/raw/raw6 /dev/disk/by-id/scsi-3600a0b800034d6790000c4348519b7-part1
raw /dev/raw/raw7 /dev/disk/by-id/scsi-3600a0b80003499160000d2b48519ea7-part1
raw /dev/raw/raw8 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part1
raw /dev/raw/raw9 /dev/disk/by-id/scsi-3600a0b80003498e00000c6c4851a121-part1
raw /dev/raw/raw10 /dev/disk/by-id/scsi-3600a0b800034d5bd0000c34485243f9-part1
raw /dev/raw/raw11 /dev/disk/by-id/scsi-3600a0b80003499080000d9348519482-part1
raw /dev/raw/raw12 /dev/disk/by-id/scsi-3600a0b80003460e90000add485193d2-part1
raw /dev/raw/raw13 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part1
raw /dev/raw/raw14 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part1
raw /dev/raw/raw15 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part1
raw /dev/raw/raw16 /dev/disk/by-id/scsi-3600a0b800034990e0000c7d4851a159-part1
raw /dev/raw/raw17 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part1
raw /dev/raw/raw18 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part1
raw /dev/raw/raw19 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part1
raw /dev/raw/raw20 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part1
raw /dev/raw/raw21 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part1
raw /dev/raw/raw22 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part1
raw /dev/raw/raw23 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part1
raw /dev/raw/raw24 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part1
raw /dev/raw/raw25 /dev/disk/by-id/scsi-3600a0b800036964400000aef4852690b-part1
raw /dev/raw/raw26 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part1
raw /dev/raw/raw27 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part1
raw /dev/raw/raw28 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part1
raw /dev/raw/raw29 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part1
raw /dev/raw/raw30 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part1
raw /dev/raw/raw31 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part1
raw /dev/raw/raw32 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part1
raw /dev/raw/raw33 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part1
raw /dev/raw/raw34 /dev/disk/by-id/scsi-3600a0b800034d5a800006bb4852410b-part1
raw /dev/raw/raw35 /dev/disk/by-id/scsi-3600a0b800036fefd0000084b48526d5c-part1
raw /dev/raw/raw36 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part1
raw /dev/raw/raw37 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part1
raw /dev/raw/raw38 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part1
raw /dev/raw/raw39 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part1
raw /dev/raw/raw40 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part1

raw /dev/raw/raw41 /dev/disk/by-id/scsi-3600a0b800024589c0000065148526746-part1
raw /dev/raw/raw42 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part1
raw /dev/raw/raw43 /dev/disk/by-id/scsi-3600a0b80002458c6000003cf48526584-part1
raw /dev/raw/raw44 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part1
raw /dev/raw/raw45 /dev/disk/by-id/scsi-3600a0b8000245d8e000008b148524fa4-part1
raw /dev/raw/raw46 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part1
raw /dev/raw/raw47 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part1
raw /dev/raw/raw48 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part1
raw /dev/raw/raw49 /dev/disk/by-id/scsi-3600a0b800034d5f8f0000061d48524df5-part1
raw /dev/raw/raw50 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part1
raw /dev/raw/raw51 /dev/disk/by-id/scsi-3600a0b80003a983400000d948526f37-part1
raw /dev/raw/raw52 /dev/disk/by-id/scsi-3600a0b80003ac5d2000006364852706d-part1
raw /dev/raw/raw53 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part1
raw /dev/raw/raw54 /dev/disk/by-id/scsi-3600a0b80003ac570000096b4852708d-part1
raw /dev/raw/raw55 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part1
raw /dev/raw/raw56 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part1
raw /dev/raw/raw57 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part1
raw /dev/raw/raw58 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part1
raw /dev/raw/raw59 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part1
raw /dev/raw/raw60 /dev/disk/by-id/scsi-3600a0b80003ca8f0000003174852648e-part1
raw /dev/raw/raw61 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part1
raw /dev/raw/raw62 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part1
raw /dev/raw/raw63 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part1
raw /dev/raw/raw64 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part1
raw /dev/raw/raw65 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part1
raw /dev/raw/raw66 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part1
raw /dev/raw/raw67 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part1
raw /dev/raw/raw68 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part1
raw /dev/raw/raw69 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part1
raw /dev/raw/raw70 /dev/disk/by-id/scsi-3600a0b80003a991e00000300485269f9-part1
raw /dev/raw/raw71 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part1
raw /dev/raw/raw72 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part1
raw /dev/raw/raw73 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part1
raw /dev/raw/raw74 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part1
raw /dev/raw/raw75 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part1

raw /dev/raw/raw76 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part1
raw /dev/raw/raw77 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part1
raw /dev/raw/raw78 /dev/disk/by-id/scsi-3600a0b80003a98d00000062448526fad-part1
raw /dev/raw/raw79 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part1
raw /dev/raw/raw80 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part1

raw /dev/raw/raw81 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part2
raw /dev/raw/raw82 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part2
raw /dev/raw/raw83 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part2
raw /dev/raw/raw84 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part2
raw /dev/raw/raw85 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part2
raw /dev/raw/raw86 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519bb7-part2
raw /dev/raw/raw87 /dev/disk/by-id/scsi-3600a0b80003499160000d2b48519ea7-part2
raw /dev/raw/raw88 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part2
raw /dev/raw/raw89 /dev/disk/by-id/scsi-3600a0b80003498e00000c6c4851a121-part2
raw /dev/raw/raw90 /dev/disk/by-id/scsi-3600a0b800034d5bd0000c34485243f9-part2
raw /dev/raw/raw91 /dev/disk/by-id/scsi-3600a0b80003499080000d9348519482-part2
raw /dev/raw/raw92 /dev/disk/by-id/scsi-3600a0b80003460e90000add485193d2-part2
raw /dev/raw/raw93 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part2
raw /dev/raw/raw94 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part2
raw /dev/raw/raw95 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part2
raw /dev/raw/raw96 /dev/disk/by-id/scsi-3600a0b800034990e0000c7d4851a159-part2
raw /dev/raw/raw97 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part2
raw /dev/raw/raw98 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part2
raw /dev/raw/raw99 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part2
raw /dev/raw/raw100 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part2
raw /dev/raw/raw101 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part2
raw /dev/raw/raw102 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part2
raw /dev/raw/raw103 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part2
raw /dev/raw/raw104 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part2
raw /dev/raw/raw105 /dev/disk/by-id/scsi-3600a0b800036964400000aef4852690b-part2
raw /dev/raw/raw106 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part2
raw /dev/raw/raw107 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part2
raw /dev/raw/raw108 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part2
raw /dev/raw/raw109 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part2

raw /dev/raw/raw110 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part2
raw /dev/raw/raw111 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part2
raw /dev/raw/raw112 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part2
raw /dev/raw/raw113 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part2
raw /dev/raw/raw114 /dev/disk/by-id/scsi-3600a0b800034d5a800006bbb48524108-part2
raw /dev/raw/raw115 /dev/disk/by-id/scsi-3600a0b800036fe0000084b48526d5c-part2
raw /dev/raw/raw116 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part2
raw /dev/raw/raw117 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part2
raw /dev/raw/raw118 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part2
raw /dev/raw/raw119 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part2
raw /dev/raw/raw120 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part2
raw /dev/raw/raw121 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part2
raw /dev/raw/raw122 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part2
raw /dev/raw/raw123 /dev/disk/by-id/scsi-3600a0b80002458c6000003c48526584-part2
raw /dev/raw/raw124 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part2
raw /dev/raw/raw125 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part2
raw /dev/raw/raw126 /dev/disk/by-id/scsi-3600a0b800024596100000404853a713-part2
raw /dev/raw/raw127 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part2
raw /dev/raw/raw128 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part2
raw /dev/raw/raw129 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524d4f5-part2
raw /dev/raw/raw130 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part2
raw /dev/raw/raw131 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part2
raw /dev/raw/raw132 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part2
raw /dev/raw/raw133 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part2
raw /dev/raw/raw134 /dev/disk/by-id/scsi-3600a0b80003ac570000096b4852708d-part2
raw /dev/raw/raw135 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part2
raw /dev/raw/raw136 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part2
raw /dev/raw/raw137 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part2
raw /dev/raw/raw138 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part2
raw /dev/raw/raw139 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part2
raw /dev/raw/raw140 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part2
raw /dev/raw/raw141 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part2
raw /dev/raw/raw142 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part2
raw /dev/raw/raw143 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part2
raw /dev/raw/raw144 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part2

raw /dev/raw/raw145 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part2
raw /dev/raw/raw146 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part2
raw /dev/raw/raw147 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part2
raw /dev/raw/raw148 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part2
raw /dev/raw/raw149 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part2
raw /dev/raw/raw150 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part2
raw /dev/raw/raw151 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part2
raw /dev/raw/raw152 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part2
raw /dev/raw/raw153 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part2
raw /dev/raw/raw154 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part2
raw /dev/raw/raw155 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part2
raw /dev/raw/raw156 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part2
raw /dev/raw/raw157 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part2
raw /dev/raw/raw158 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part2
raw /dev/raw/raw159 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000006648527035-part2
raw /dev/raw/raw160 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part2

raw /dev/raw/raw161 /dev/disk/by-id/scsi-3600a0b800034d63d0000003b48519090-part3
raw /dev/raw/raw162 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part3
raw /dev/raw/raw163 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part3
raw /dev/raw/raw164 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part3
raw /dev/raw/raw165 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part3
raw /dev/raw/raw166 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519bb7-part3
raw /dev/raw/raw167 /dev/disk/by-id/scsi-3600a0b800034991600000d2b48519ea7-part3
raw /dev/raw/raw168 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part3
raw /dev/raw/raw169 /dev/disk/by-id/scsi-3600a0b80003498e00000c6c4851a121-part3
raw /dev/raw/raw170 /dev/disk/by-id/scsi-3600a0b800034d5db00000c34485243f9-part3
raw /dev/raw/raw171 /dev/disk/by-id/scsi-3600a0b80003499080000009348519482-part3
raw /dev/raw/raw172 /dev/disk/by-id/scsi-3600a0b80003460e900000add485193d2-part3
raw /dev/raw/raw173 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part3
raw /dev/raw/raw174 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part3
raw /dev/raw/raw175 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part3
raw /dev/raw/raw176 /dev/disk/by-id/scsi-3600a0b800034990e00000c7d4851a159-part3
raw /dev/raw/raw177 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part3
raw /dev/raw/raw178 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part3

raw /dev/raw/raw179 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part3
raw /dev/raw/raw180 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part3
raw /dev/raw/raw181 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part3
raw /dev/raw/raw182 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part3
raw /dev/raw/raw183 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part3
raw /dev/raw/raw184 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part3
raw /dev/raw/raw185 /dev/disk/by-id/scsi-3600a0b800036964400000aef485269b0-part3
raw /dev/raw/raw186 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part3
raw /dev/raw/raw187 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part3
raw /dev/raw/raw188 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part3
raw /dev/raw/raw189 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part3
raw /dev/raw/raw190 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part3
raw /dev/raw/raw191 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part3
raw /dev/raw/raw192 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part3
raw /dev/raw/raw193 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part3
raw /dev/raw/raw194 /dev/disk/by-id/scsi-3600a0b800034d5a800006bbb48524108-part3
raw /dev/raw/raw195 /dev/disk/by-id/scsi-3600a0b800036fe0000084b48526d5c-part3
raw /dev/raw/raw196 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part3
raw /dev/raw/raw197 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part3
raw /dev/raw/raw198 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part3
raw /dev/raw/raw199 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part3
raw /dev/raw/raw200 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part3
raw /dev/raw/raw201 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part3
raw /dev/raw/raw202 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part3
raw /dev/raw/raw203 /dev/disk/by-id/scsi-3600a0b80002458c6000003c48526584-part3
raw /dev/raw/raw204 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part3
raw /dev/raw/raw205 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part3
raw /dev/raw/raw206 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part3
raw /dev/raw/raw207 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part3
raw /dev/raw/raw208 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part3
raw /dev/raw/raw209 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part3
raw /dev/raw/raw210 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part3
raw /dev/raw/raw211 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part3
raw /dev/raw/raw212 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part3
raw /dev/raw/raw213 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part3

raw /dev/raw/raw214 /dev/disk/by-id/scsi-3600a0b80003ac570000096b4852708d-part3
raw /dev/raw/raw215 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part3
raw /dev/raw/raw216 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part3
raw /dev/raw/raw217 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part3
raw /dev/raw/raw218 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part3
raw /dev/raw/raw219 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part3
raw /dev/raw/raw220 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part3
raw /dev/raw/raw221 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part3
raw /dev/raw/raw222 /dev/disk/by-id/scsi-3600a0b80003ac21c000006cf4852657a-part3
raw /dev/raw/raw223 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part3
raw /dev/raw/raw224 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part3
raw /dev/raw/raw225 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part3
raw /dev/raw/raw226 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part3
raw /dev/raw/raw227 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part3
raw /dev/raw/raw228 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part3
raw /dev/raw/raw229 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part3
raw /dev/raw/raw230 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part3
raw /dev/raw/raw231 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part3
raw /dev/raw/raw232 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part3
raw /dev/raw/raw233 /dev/disk/by-id/scsi-3600a0b80003ac60b000007004852682e-part3
raw /dev/raw/raw234 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part3
raw /dev/raw/raw235 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part3
raw /dev/raw/raw236 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part3
raw /dev/raw/raw237 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part3
raw /dev/raw/raw238 /dev/disk/by-id/scsi-3600a0b80003a98d00000062448526fad-part3
raw /dev/raw/raw239 /dev/disk/by-id/scsi-3600a0b80003a9a2f00000066e48527035-part3
raw /dev/raw/raw240 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part3

raw /dev/raw/raw241 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part5
raw /dev/raw/raw242 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part5
raw /dev/raw/raw243 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part5
raw /dev/raw/raw244 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part5
raw /dev/raw/raw245 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part5
raw /dev/raw/raw246 /dev/disk/by-id/scsi-3600a0b800034d67900000c3448519bb7-part5
raw /dev/raw/raw247 /dev/disk/by-id/scsi-3600a0b800034991600000d2b48519ea7-part5

raw /dev/raw/raw248 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part5
raw /dev/raw/raw249 /dev/disk/by-id/scsi-3600a0b80003498e000000c6c4851a121-part5
raw /dev/raw/raw250 /dev/disk/by-id/scsi-3600a0b800034d5db00000c34485243f9-part5
raw /dev/raw/raw251 /dev/disk/by-id/scsi-3600a0b800034990800000d9348519482-part5
raw /dev/raw/raw252 /dev/disk/by-id/scsi-3600a0b80003460e900000ad485193d2-part5
raw /dev/raw/raw253 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part5
raw /dev/raw/raw254 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part5
raw /dev/raw/raw255 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part5
raw /dev/raw/raw256 /dev/disk/by-id/scsi-3600a0b800034990e00000c7d4851a159-part5
raw /dev/raw/raw257 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part5
raw /dev/raw/raw258 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part5
raw /dev/raw/raw259 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part5
raw /dev/raw/raw260 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part5
raw /dev/raw/raw261 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part5
raw /dev/raw/raw262 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part5
raw /dev/raw/raw263 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part5
raw /dev/raw/raw264 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part5
raw /dev/raw/raw265 /dev/disk/by-id/scsi-3600a0b800036964400000ae485269b0-part5
raw /dev/raw/raw266 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part5
raw /dev/raw/raw267 /dev/disk/by-id/scsi-3600a0b8000369884000005a44852768d-part5
raw /dev/raw/raw268 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part5
raw /dev/raw/raw269 /dev/disk/by-id/scsi-3600a0b800034d6db00000697485249a5-part5
raw /dev/raw/raw270 /dev/disk/by-id/scsi-3600a0b8000369e2d00000087e485275b5-part5
raw /dev/raw/raw271 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part5
raw /dev/raw/raw272 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part5
raw /dev/raw/raw273 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part5
raw /dev/raw/raw274 /dev/disk/by-id/scsi-3600a0b800034d5a800006bbb48524108-part5
raw /dev/raw/raw275 /dev/disk/by-id/scsi-3600a0b800036fef0000084b48526d5c-part5
raw /dev/raw/raw276 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part5
raw /dev/raw/raw277 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part5
raw /dev/raw/raw278 /dev/disk/by-id/scsi-3600a0b800034d6db0000052248524b27-part5
raw /dev/raw/raw279 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part5
raw /dev/raw/raw280 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part5
raw /dev/raw/raw281 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part5
raw /dev/raw/raw282 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part5

raw /dev/raw/raw283 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part5
raw /dev/raw/raw284 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part5
raw /dev/raw/raw285 /dev/disk/by-id/scsi-3600a0b8000245d6b000008b148524fa4-part5
raw /dev/raw/raw286 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part5
raw /dev/raw/raw287 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part5
raw /dev/raw/raw288 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part5
raw /dev/raw/raw289 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part5
raw /dev/raw/raw290 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part5
raw /dev/raw/raw291 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part5
raw /dev/raw/raw292 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part5
raw /dev/raw/raw293 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part5
raw /dev/raw/raw294 /dev/disk/by-id/scsi-3600a0b80003ac570000096b4852708d-part5
raw /dev/raw/raw295 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part5
raw /dev/raw/raw296 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part5
raw /dev/raw/raw297 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part5
raw /dev/raw/raw298 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part5
raw /dev/raw/raw299 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part5
raw /dev/raw/raw300 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part5
raw /dev/raw/raw301 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part5
raw /dev/raw/raw302 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part5
raw /dev/raw/raw303 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part5
raw /dev/raw/raw304 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part5
raw /dev/raw/raw305 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part5
raw /dev/raw/raw306 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part5
raw /dev/raw/raw307 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part5
raw /dev/raw/raw308 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part5
raw /dev/raw/raw309 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part5
raw /dev/raw/raw310 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part5
raw /dev/raw/raw311 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part5
raw /dev/raw/raw312 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part5
raw /dev/raw/raw313 /dev/disk/by-id/scsi-3600a0b80003ac60b000007004852682e-part5
raw /dev/raw/raw314 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part5
raw /dev/raw/raw315 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part5
raw /dev/raw/raw316 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part5
raw /dev/raw/raw317 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part5

raw /dev/raw/raw318 /dev/disk/by-id/scsi-3600a0b80003a99d0000062448526fad-part5
raw /dev/raw/raw319 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part5
raw /dev/raw/raw320 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part5

raw /dev/raw/raw321 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b4851909c-part6
raw /dev/raw/raw322 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part6
raw /dev/raw/raw323 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part6
raw /dev/raw/raw324 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part6
raw /dev/raw/raw325 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part6
raw /dev/raw/raw326 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519bb7-part6
raw /dev/raw/raw327 /dev/disk/by-id/scsi-3600a0b80003499160000d2b48519ea7-part6
raw /dev/raw/raw328 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part6
raw /dev/raw/raw329 /dev/disk/by-id/scsi-3600a0b80003498e000000c6c4851a121-part6
raw /dev/raw/raw330 /dev/disk/by-id/scsi-3600a0b800034d5b00000c34485243f9-part6
raw /dev/raw/raw331 /dev/disk/by-id/scsi-3600a0b80003499080000d9348519482-part6
raw /dev/raw/raw332 /dev/disk/by-id/scsi-3600a0b80003460e900000add485193d2-part6
raw /dev/raw/raw333 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part6
raw /dev/raw/raw334 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part6
raw /dev/raw/raw335 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part6
raw /dev/raw/raw336 /dev/disk/by-id/scsi-3600a0b800034990e00000c7d4851a159-part6
raw /dev/raw/raw337 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part6
raw /dev/raw/raw338 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part6
raw /dev/raw/raw339 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part6
raw /dev/raw/raw340 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part6
raw /dev/raw/raw341 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part6
raw /dev/raw/raw342 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part6
raw /dev/raw/raw343 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part6
raw /dev/raw/raw344 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part6
raw /dev/raw/raw345 /dev/disk/by-id/scsi-3600a0b800036964400000aef485269b0-part6
raw /dev/raw/raw346 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part6
raw /dev/raw/raw347 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part6
raw /dev/raw/raw348 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part6
raw /dev/raw/raw349 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part6
raw /dev/raw/raw350 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part6
raw /dev/raw/raw351 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part6

raw /dev/raw/raw352 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part6
raw /dev/raw/raw353 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part6
raw /dev/raw/raw354 /dev/disk/by-id/scsi-3600a0b800034d5a800006bb48524108-part6
raw /dev/raw/raw355 /dev/disk/by-id/scsi-3600a0b800036fefdd0000084b48526d5c-part6
raw /dev/raw/raw356 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part6
raw /dev/raw/raw357 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part6
raw /dev/raw/raw358 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part6
raw /dev/raw/raw359 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part6
raw /dev/raw/raw360 /dev/disk/by-id/scsi-3600a0b800034d6c5000007948524b63-part6
raw /dev/raw/raw361 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part6
raw /dev/raw/raw362 /dev/disk/by-id/scsi-3600a0b800034d720000068448524232-part6
raw /dev/raw/raw363 /dev/disk/by-id/scsi-3600a0b80002458c6000003c48526584-part6
raw /dev/raw/raw364 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part6
raw /dev/raw/raw365 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part6
raw /dev/raw/raw366 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part6
raw /dev/raw/raw367 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part6
raw /dev/raw/raw368 /dev/disk/by-id/scsi-3600a0b800024589a0000042485273e7-part6
raw /dev/raw/raw369 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part6
raw /dev/raw/raw370 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part6
raw /dev/raw/raw371 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part6
raw /dev/raw/raw372 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part6
raw /dev/raw/raw373 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part6
raw /dev/raw/raw374 /dev/disk/by-id/scsi-3600a0b80003ac570000096b4852708d-part6
raw /dev/raw/raw375 /dev/disk/by-id/scsi-3600a0b800039c6f0000033148526e53-part6
raw /dev/raw/raw376 /dev/disk/by-id/scsi-3600a0b80003ac5c60000079a485262f2-part6
raw /dev/raw/raw377 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part6
raw /dev/raw/raw378 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part6
raw /dev/raw/raw379 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part6
raw /dev/raw/raw380 /dev/disk/by-id/scsi-3600a0b800039ca8f000003174852648e-part6
raw /dev/raw/raw381 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part6
raw /dev/raw/raw382 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part6
raw /dev/raw/raw383 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part6
raw /dev/raw/raw384 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part6
raw /dev/raw/raw385 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part6
raw /dev/raw/raw386 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part6

raw /dev/raw/raw387 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part6
raw /dev/raw/raw388 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part6
raw /dev/raw/raw389 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part6
raw /dev/raw/raw390 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part6
raw /dev/raw/raw391 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part6
raw /dev/raw/raw392 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part6
raw /dev/raw/raw393 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part6
raw /dev/raw/raw394 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part6
raw /dev/raw/raw395 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part6
raw /dev/raw/raw396 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part6
raw /dev/raw/raw397 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part6
raw /dev/raw/raw398 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part6
raw /dev/raw/raw399 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part6
raw /dev/raw/raw400 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part6

raw /dev/raw/raw401 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part7
raw /dev/raw/raw402 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part7
raw /dev/raw/raw403 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part7
raw /dev/raw/raw404 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part7
raw /dev/raw/raw405 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part7
raw /dev/raw/raw406 /dev/disk/by-id/scsi-3600a0b800034d67900000c6c4851a121-part7
raw /dev/raw/raw407 /dev/disk/by-id/scsi-3600a0b80003499160000d2b48519ea7-part7
raw /dev/raw/raw408 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part7
raw /dev/raw/raw409 /dev/disk/by-id/scsi-3600a0b80003498e000000c6c4851a121-part7
raw /dev/raw/raw410 /dev/disk/by-id/scsi-3600a0b800034d5b00000c34485243f9-part7
raw /dev/raw/raw411 /dev/disk/by-id/scsi-3600a0b80003499080000d9348519482-part7
raw /dev/raw/raw412 /dev/disk/by-id/scsi-3600a0b80003460e900000add485193d2-part7
raw /dev/raw/raw413 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part7
raw /dev/raw/raw414 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part7
raw /dev/raw/raw415 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part7
raw /dev/raw/raw416 /dev/disk/by-id/scsi-3600a0b800034990e00000c7d4851a159-part7
raw /dev/raw/raw417 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part7
raw /dev/raw/raw418 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part7
raw /dev/raw/raw419 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part7
raw /dev/raw/raw420 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part7

raw /dev/raw/raw421 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part7
raw /dev/raw/raw422 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part7
raw /dev/raw/raw423 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part7
raw /dev/raw/raw424 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part7
raw /dev/raw/raw425 /dev/disk/by-id/scsi-3600a0b800036964400000aef485269b0-part7
raw /dev/raw/raw426 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part7
raw /dev/raw/raw427 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part7
raw /dev/raw/raw428 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part7
raw /dev/raw/raw429 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part7
raw /dev/raw/raw430 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part7
raw /dev/raw/raw431 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523f00-part7
raw /dev/raw/raw432 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part7
raw /dev/raw/raw433 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part7
raw /dev/raw/raw434 /dev/disk/by-id/scsi-3600a0b800034d5a800006bb48524108-part7
raw /dev/raw/raw435 /dev/disk/by-id/scsi-3600a0b800036fefd0000084b48526d5c-part7
raw /dev/raw/raw436 /dev/disk/by-id/scsi-3600a0b800034608000010c948524c29-part7
raw /dev/raw/raw437 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part7
raw /dev/raw/raw438 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part7
raw /dev/raw/raw439 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part7
raw /dev/raw/raw440 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b3-part7
raw /dev/raw/raw441 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part7
raw /dev/raw/raw442 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part7
raw /dev/raw/raw443 /dev/disk/by-id/scsi-3600a0b80002458c6000003cf48526584-part7
raw /dev/raw/raw444 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part7
raw /dev/raw/raw445 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part7
raw /dev/raw/raw446 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part7
raw /dev/raw/raw447 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part7
raw /dev/raw/raw448 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part7
raw /dev/raw/raw449 /dev/disk/by-id/scsi-3600a0b800034d58f0000061048524df5-part7
raw /dev/raw/raw450 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part7
raw /dev/raw/raw451 /dev/disk/by-id/scsi-3600a0b80003a9834000006948526f37-part7
raw /dev/raw/raw452 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part7
raw /dev/raw/raw453 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part7
raw /dev/raw/raw454 /dev/disk/by-id/scsi-3600a0b80003ac5700000096b4852708d-part7
raw /dev/raw/raw455 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part7

raw /dev/raw/raw456 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part7
raw /dev/raw/raw457 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part7
raw /dev/raw/raw458 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part7
raw /dev/raw/raw459 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part7
raw /dev/raw/raw460 /dev/disk/by-id/scsi-3600a0b800039ca8000003174852648e-part7
raw /dev/raw/raw461 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part7
raw /dev/raw/raw462 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part7
raw /dev/raw/raw463 /dev/disk/by-id/scsi-3600a0b80003a9a800000815485266e8-part7
raw /dev/raw/raw464 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part7
raw /dev/raw/raw465 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part7
raw /dev/raw/raw466 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part7
raw /dev/raw/raw467 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part7
raw /dev/raw/raw468 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part7
raw /dev/raw/raw469 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part7
raw /dev/raw/raw470 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part7
raw /dev/raw/raw471 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part7
raw /dev/raw/raw472 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part7
raw /dev/raw/raw473 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part7
raw /dev/raw/raw474 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part7
raw /dev/raw/raw475 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part7
raw /dev/raw/raw476 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part7
raw /dev/raw/raw477 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part7
raw /dev/raw/raw478 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part7
raw /dev/raw/raw479 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part7
raw /dev/raw/raw480 /dev/disk/by-id/scsi-3600a0b80003ac56b0000030648527075-part7
raw /dev/raw/raw481 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part8
raw /dev/raw/raw482 /dev/disk/by-id/scsi-3600a0b800034999f0000112485190fc-part8
raw /dev/raw/raw483 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part8
raw /dev/raw/raw484 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part8
raw /dev/raw/raw485 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part8
raw /dev/raw/raw486 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519b7-part8
raw /dev/raw/raw487 /dev/disk/by-id/scsi-3600a0b8000349916000002b48519ea7-part8
raw /dev/raw/raw488 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part8
raw /dev/raw/raw489 /dev/disk/by-id/scsi-3600a0b80003498e000000c6c4851a121-part8

raw /dev/raw/raw490 /dev/disk/by-id/scsi-3600a0b800034d5bd00000c34485243f9-part8
raw /dev/raw/raw491 /dev/disk/by-id/scsi-3600a0b80003499800000d9348519482-part8
raw /dev/raw/raw492 /dev/disk/by-id/scsi-3600a0b80003460e900000add485193d2-part8
raw /dev/raw/raw493 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part8
raw /dev/raw/raw494 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part8
raw /dev/raw/raw495 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part8
raw /dev/raw/raw496 /dev/disk/by-id/scsi-3600a0b800034990e00000c7d4851a159-part8
raw /dev/raw/raw497 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part8
raw /dev/raw/raw498 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part8
raw /dev/raw/raw499 /dev/disk/by-id/scsi-3600a0b800034999d00000816485247f5-part8
raw /dev/raw/raw500 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part8
raw /dev/raw/raw501 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part8
raw /dev/raw/raw502 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part8
raw /dev/raw/raw503 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part8
raw /dev/raw/raw504 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part8
raw /dev/raw/raw505 /dev/disk/by-id/scsi-3600a0b800036964400000aef485269b0-part8
raw /dev/raw/raw506 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part8
raw /dev/raw/raw507 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part8
raw /dev/raw/raw508 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part8
raw /dev/raw/raw509 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part8
raw /dev/raw/raw510 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part8
raw /dev/raw/raw511 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part8
raw /dev/raw/raw512 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part8
raw /dev/raw/raw513 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part8
raw /dev/raw/raw514 /dev/disk/by-id/scsi-3600a0b800034d5a800006bb48524108-part8
raw /dev/raw/raw515 /dev/disk/by-id/scsi-3600a0b800036fefd0000084b48526d5c-part8
raw /dev/raw/raw516 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part8
raw /dev/raw/raw517 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part8
raw /dev/raw/raw518 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part8
raw /dev/raw/raw519 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part8
raw /dev/raw/raw520 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part8
raw /dev/raw/raw521 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part8
raw /dev/raw/raw522 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part8
raw /dev/raw/raw523 /dev/disk/by-id/scsi-3600a0b80002458c6000003cf48526584-part8
raw /dev/raw/raw524 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part8

raw /dev/raw/raw525 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part8
raw /dev/raw/raw526 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part8
raw /dev/raw/raw527 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part8
raw /dev/raw/raw528 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part8
raw /dev/raw/raw529 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part8
raw /dev/raw/raw530 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part8
raw /dev/raw/raw531 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part8
raw /dev/raw/raw532 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part8
raw /dev/raw/raw533 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part8
raw /dev/raw/raw534 /dev/disk/by-id/scsi-3600a0b80003ac5700000096b4852708d-part8
raw /dev/raw/raw535 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part8
raw /dev/raw/raw536 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part8
raw /dev/raw/raw537 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part8
raw /dev/raw/raw538 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part8
raw /dev/raw/raw539 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part8
raw /dev/raw/raw540 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part8
raw /dev/raw/raw541 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part8
raw /dev/raw/raw542 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part8
raw /dev/raw/raw543 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part8
raw /dev/raw/raw544 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part8
raw /dev/raw/raw545 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part8
raw /dev/raw/raw546 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part8
raw /dev/raw/raw547 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part8
raw /dev/raw/raw548 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part8
raw /dev/raw/raw549 /dev/disk/by-id/scsi-3600a0b80003ac64b000007944852706f5-part8
raw /dev/raw/raw550 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part8
raw /dev/raw/raw551 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part8
raw /dev/raw/raw552 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part8
raw /dev/raw/raw553 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part8
raw /dev/raw/raw554 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part8
raw /dev/raw/raw555 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part8
raw /dev/raw/raw556 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part8
raw /dev/raw/raw557 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part8
raw /dev/raw/raw558 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part8
raw /dev/raw/raw559 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part8

raw /dev/raw/raw560 /dev/disk/by-id/scsi-3600a0b80003ac56b00000030648527075-part8
raw /dev/raw/raw561 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part9
raw /dev/raw/raw562 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part9
raw /dev/raw/raw563 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part9
raw /dev/raw/raw564 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part9
raw /dev/raw/raw565 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part9
raw /dev/raw/raw566 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519bb7-part9
raw /dev/raw/raw567 /dev/disk/by-id/scsi-3600a0b80003499160000d2b48519ea7-part9
raw /dev/raw/raw568 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part9
raw /dev/raw/raw569 /dev/disk/by-id/scsi-3600a0b80003498e000000c6c4851a121-part9
raw /dev/raw/raw570 /dev/disk/by-id/scsi-3600a0b800034d5b00000c34485243f9-part9
raw /dev/raw/raw571 /dev/disk/by-id/scsi-3600a0b800034990800000d9348519482-part9
raw /dev/raw/raw572 /dev/disk/by-id/scsi-3600a0b80003460e900000add485193d2-part9
raw /dev/raw/raw573 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part9
raw /dev/raw/raw574 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part9
raw /dev/raw/raw575 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part9
raw /dev/raw/raw576 /dev/disk/by-id/scsi-3600a0b800034990e00000c7d4851a159-part9
raw /dev/raw/raw577 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part9
raw /dev/raw/raw578 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part9
raw /dev/raw/raw579 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part9
raw /dev/raw/raw580 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part9
raw /dev/raw/raw581 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part9
raw /dev/raw/raw582 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part9
raw /dev/raw/raw583 /dev/disk/by-id/scsi-3600a0b800036945800000534852694c-part9
raw /dev/raw/raw584 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part9
raw /dev/raw/raw585 /dev/disk/by-id/scsi-3600a0b800036964400000ae4f485269b0-part9
raw /dev/raw/raw586 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part9
raw /dev/raw/raw587 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part9
raw /dev/raw/raw588 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part9
raw /dev/raw/raw589 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part9
raw /dev/raw/raw590 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part9
raw /dev/raw/raw591 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part9
raw /dev/raw/raw592 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part9
raw /dev/raw/raw593 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part9

raw /dev/raw/raw594 /dev/disk/by-id/scsi-3600a0b800034d5a800006bbb48524108-part9
raw /dev/raw/raw595 /dev/disk/by-id/scsi-3600a0b800036fef0000084b48526d5c-part9
raw /dev/raw/raw596 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part9
raw /dev/raw/raw597 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part9
raw /dev/raw/raw598 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part9
raw /dev/raw/raw599 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part9
raw /dev/raw/raw600 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part9
raw /dev/raw/raw601 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part9
raw /dev/raw/raw602 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part9
raw /dev/raw/raw603 /dev/disk/by-id/scsi-3600a0b80002458c6000003cf48526584-part9
raw /dev/raw/raw604 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part9
raw /dev/raw/raw605 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part9
raw /dev/raw/raw606 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part9
raw /dev/raw/raw607 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part9
raw /dev/raw/raw608 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part9
raw /dev/raw/raw609 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part9
raw /dev/raw/raw610 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part9
raw /dev/raw/raw611 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part9
raw /dev/raw/raw612 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part9
raw /dev/raw/raw613 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part9
raw /dev/raw/raw614 /dev/disk/by-id/scsi-3600a0b80003ac5700000096b4852708d-part9
raw /dev/raw/raw615 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part9
raw /dev/raw/raw616 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part9
raw /dev/raw/raw617 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part9
raw /dev/raw/raw618 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part9
raw /dev/raw/raw619 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part9
raw /dev/raw/raw620 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part9
raw /dev/raw/raw621 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part9
raw /dev/raw/raw622 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part9
raw /dev/raw/raw623 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part9
raw /dev/raw/raw624 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part9
raw /dev/raw/raw625 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part9
raw /dev/raw/raw626 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part9
raw /dev/raw/raw627 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526fad-part9
raw /dev/raw/raw628 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part9

raw /dev/raw/raw629 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part9
raw /dev/raw/raw630 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part9
raw /dev/raw/raw631 /dev/disk/by-id/scsi-3600a0b80003ac6250000084348527627e-part9
raw /dev/raw/raw632 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part9
raw /dev/raw/raw633 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part9
raw /dev/raw/raw634 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part9
raw /dev/raw/raw635 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part9
raw /dev/raw/raw636 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part9
raw /dev/raw/raw637 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part9
raw /dev/raw/raw638 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part9
raw /dev/raw/raw639 /dev/disk/by-id/scsi-3600a0b80003a9a20000066e48527035-part9
raw /dev/raw/raw640 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part9

raw /dev/raw/raw641 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part10
raw /dev/raw/raw642 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part10
raw /dev/raw/raw643 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part10
raw /dev/raw/raw644 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part10
raw /dev/raw/raw645 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part10
raw /dev/raw/raw646 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519bb7-part10
raw /dev/raw/raw647 /dev/disk/by-id/scsi-3600a0b800034991600000d2b48519ea7-part10
raw /dev/raw/raw648 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part10
raw /dev/raw/raw649 /dev/disk/by-id/scsi-3600a0b80003498e00000c6c4851a121-part10
raw /dev/raw/raw650 /dev/disk/by-id/scsi-3600a0b800034d5b00000c34485243f9-part10
raw /dev/raw/raw651 /dev/disk/by-id/scsi-3600a0b80003499080000d9348519482-part10
raw /dev/raw/raw652 /dev/disk/by-id/scsi-3600a0b80003460e90000dadd485193d2-part10
raw /dev/raw/raw653 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part10
raw /dev/raw/raw654 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part10
raw /dev/raw/raw655 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part10
raw /dev/raw/raw656 /dev/disk/by-id/scsi-3600a0b800034990e00000c7d4851a159-part10
raw /dev/raw/raw657 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part10
raw /dev/raw/raw658 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part10
raw /dev/raw/raw659 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part10
raw /dev/raw/raw660 /dev/disk/by-id/scsi-3600a0b8000349910000011c648524411-part10
raw /dev/raw/raw661 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part10
raw /dev/raw/raw662 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part10

raw /dev/raw/raw663 /dev/disk/by-id/scsi-3600a0b80003696458000005314852694c-part10
raw /dev/raw/raw664 /dev/disk/by-id/scsi-3600a0b800034d6cd000007c48523cf8-part10
raw /dev/raw/raw665 /dev/disk/by-id/scsi-3600a0b800036964400000ae485269b0-part10
raw /dev/raw/raw666 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part10
raw /dev/raw/raw667 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part10
raw /dev/raw/raw668 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part10
raw /dev/raw/raw669 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part10
raw /dev/raw/raw670 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part10
raw /dev/raw/raw671 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part10
raw /dev/raw/raw672 /dev/disk/by-id/scsi-3600a0b80002458980000046485265bc-part10
raw /dev/raw/raw673 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part10
raw /dev/raw/raw674 /dev/disk/by-id/scsi-3600a0b800034d5a800006bb48524108-part10
raw /dev/raw/raw675 /dev/disk/by-id/scsi-3600a0b800036fef0000084b48526d5c-part10
raw /dev/raw/raw676 /dev/disk/by-id/scsi-3600a0b800034608d000010c48524c29-part10
raw /dev/raw/raw677 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part10
raw /dev/raw/raw678 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part10
raw /dev/raw/raw679 /dev/disk/by-id/scsi-3600a0b800034d7b00000069a48524d0d-part10
raw /dev/raw/raw680 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part10
raw /dev/raw/raw681 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part10
raw /dev/raw/raw682 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part10
raw /dev/raw/raw683 /dev/disk/by-id/scsi-3600a0b80002458c6000003c48526584-part10
raw /dev/raw/raw684 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part10
raw /dev/raw/raw685 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part10
raw /dev/raw/raw686 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part10
raw /dev/raw/raw687 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part10
raw /dev/raw/raw688 /dev/disk/by-id/scsi-3600a0b800024589a00000421485273e7-part10
raw /dev/raw/raw689 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part10
raw /dev/raw/raw690 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part10
raw /dev/raw/raw691 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part10
raw /dev/raw/raw692 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part10
raw /dev/raw/raw693 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part10
raw /dev/raw/raw694 /dev/disk/by-id/scsi-3600a0b80003ac570000096b4852708d-part10
raw /dev/raw/raw695 /dev/disk/by-id/scsi-3600a0b800039c6f00000033148526e53-part10
raw /dev/raw/raw696 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part10
raw /dev/raw/raw697 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part10

raw /dev/raw/raw698 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part10
raw /dev/raw/raw699 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part10
raw /dev/raw/raw700 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part10
raw /dev/raw/raw701 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part10
raw /dev/raw/raw702 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part10
raw /dev/raw/raw703 /dev/disk/by-id/scsi-3600a0b80003a9a800000815485266e8-part10
raw /dev/raw/raw704 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part10
raw /dev/raw/raw705 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part10
raw /dev/raw/raw706 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part10
raw /dev/raw/raw707 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part10
raw /dev/raw/raw708 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part10
raw /dev/raw/raw709 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part10
raw /dev/raw/raw710 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part10
raw /dev/raw/raw711 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part10
raw /dev/raw/raw712 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part10
raw /dev/raw/raw713 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part10
raw /dev/raw/raw714 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part10
raw /dev/raw/raw715 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part10
raw /dev/raw/raw716 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part10
raw /dev/raw/raw717 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part10
raw /dev/raw/raw718 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part10
raw /dev/raw/raw719 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part10
raw /dev/raw/raw720 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part10

raw /dev/raw/raw721 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part11
raw /dev/raw/raw722 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part11
raw /dev/raw/raw723 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part11
raw /dev/raw/raw724 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part11
raw /dev/raw/raw725 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part11
raw /dev/raw/raw726 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519bb7-part11
raw /dev/raw/raw727 /dev/disk/by-id/scsi-3600a0b800034991600000d2b48519ea7-part11
raw /dev/raw/raw728 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part11
raw /dev/raw/raw729 /dev/disk/by-id/scsi-3600a0b80003498e00000c6c4851a121-part11
raw /dev/raw/raw730 /dev/disk/by-id/scsi-3600a0b800034d5b00000c34485243f9-part11
raw /dev/raw/raw731 /dev/disk/by-id/scsi-3600a0b80003499080000d9348519482-part11

raw /dev/raw/raw732 /dev/disk/by-id/scsi-3600a0b80003460e90000dadd485193d2-part11
raw /dev/raw/raw733 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part11
raw /dev/raw/raw734 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part11
raw /dev/raw/raw735 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part11
raw /dev/raw/raw736 /dev/disk/by-id/scsi-3600a0b800034990e0000c7d4851a159-part11
raw /dev/raw/raw737 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part11
raw /dev/raw/raw738 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part11
raw /dev/raw/raw739 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part11
raw /dev/raw/raw740 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part11
raw /dev/raw/raw741 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part11
raw /dev/raw/raw742 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part11
raw /dev/raw/raw743 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part11
raw /dev/raw/raw744 /dev/disk/by-id/scsi-3600a0b800034d6cc000007fc48523cf8-part11
raw /dev/raw/raw745 /dev/disk/by-id/scsi-3600a0b800036964400000aef485269b0-part11
raw /dev/raw/raw746 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part11
raw /dev/raw/raw747 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part11
raw /dev/raw/raw748 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part11
raw /dev/raw/raw749 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part11
raw /dev/raw/raw750 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part11
raw /dev/raw/raw751 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part11
raw /dev/raw/raw752 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part11
raw /dev/raw/raw753 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part11
raw /dev/raw/raw754 /dev/disk/by-id/scsi-3600a0b800034d5a800006bb48524108-part11
raw /dev/raw/raw755 /dev/disk/by-id/scsi-3600a0b800036f6fd0000084b48526d5c-part11
raw /dev/raw/raw756 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part11
raw /dev/raw/raw757 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part11
raw /dev/raw/raw758 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part11
raw /dev/raw/raw759 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part11
raw /dev/raw/raw760 /dev/disk/by-id/scsi-3600a0b800034d6c50000079148524b33-part11
raw /dev/raw/raw761 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part11
raw /dev/raw/raw762 /dev/disk/by-id/scsi-3600a0b800034d7620000068448524232-part11
raw /dev/raw/raw763 /dev/disk/by-id/scsi-3600a0b80002458c6000003cf48526584-part11
raw /dev/raw/raw764 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part11
raw /dev/raw/raw765 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part11
raw /dev/raw/raw766 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part11

raw /dev/raw/raw767 /dev/disk/by-id/scsi-3600a0b800034659d000006ad48524d37-part11
raw /dev/raw/raw768 /dev/disk/by-id/scsi-3600a0b800024589a00000421485273e7-part11
raw /dev/raw/raw769 /dev/disk/by-id/scsi-3600a0b800034658f0000061d48524df5-part11
raw /dev/raw/raw770 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part11
raw /dev/raw/raw771 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part11
raw /dev/raw/raw772 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part11
raw /dev/raw/raw773 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part11
raw /dev/raw/raw774 /dev/disk/by-id/scsi-3600a0b80003ac5700000096b4852708d-part11
raw /dev/raw/raw775 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part11
raw /dev/raw/raw776 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part11
raw /dev/raw/raw777 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part11
raw /dev/raw/raw778 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part11
raw /dev/raw/raw779 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part11
raw /dev/raw/raw780 /dev/disk/by-id/scsi-3600a0b800039ca8000003174852648e-part11
raw /dev/raw/raw781 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part11
raw /dev/raw/raw782 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part11
raw /dev/raw/raw783 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part11
raw /dev/raw/raw784 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part11
raw /dev/raw/raw785 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part11
raw /dev/raw/raw786 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part11
raw /dev/raw/raw787 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part11
raw /dev/raw/raw788 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part11
raw /dev/raw/raw789 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part11
raw /dev/raw/raw790 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part11
raw /dev/raw/raw791 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part11
raw /dev/raw/raw792 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part11
raw /dev/raw/raw793 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part11
raw /dev/raw/raw794 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part11
raw /dev/raw/raw795 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part11
raw /dev/raw/raw796 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part11
raw /dev/raw/raw797 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part11
raw /dev/raw/raw798 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part11
raw /dev/raw/raw799 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part11
raw /dev/raw/raw800 /dev/disk/by-id/scsi-3600a0b80003ac56b0000030648527075-part11

raw /dev/raw/raw801 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part12
raw /dev/raw/raw802 /dev/disk/by-id/scsi-3600a0b800034990f000011f2485190fc-part12
raw /dev/raw/raw803 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part12
raw /dev/raw/raw804 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part12
raw /dev/raw/raw805 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part12
raw /dev/raw/raw806 /dev/disk/by-id/scsi-3600a0b800034d67900000c4348519bb7-part12
raw /dev/raw/raw807 /dev/disk/by-id/scsi-3600a0b800034991600000d2b48519ea7-part12
raw /dev/raw/raw808 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part12
raw /dev/raw/raw809 /dev/disk/by-id/scsi-3600a0b80003498e000000c6c4851a121-part12
raw /dev/raw/raw810 /dev/disk/by-id/scsi-3600a0b800034d5bd00000c34485243f9-part12
raw /dev/raw/raw811 /dev/disk/by-id/scsi-3600a0b800034990800000d9348519482-part12
raw /dev/raw/raw812 /dev/disk/by-id/scsi-3600a0b80003460e90000add485193d2-part12
raw /dev/raw/raw813 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part12
raw /dev/raw/raw814 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part12
raw /dev/raw/raw815 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part12
raw /dev/raw/raw816 /dev/disk/by-id/scsi-3600a0b800034990e0000c7d4851a159-part12
raw /dev/raw/raw817 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part12
raw /dev/raw/raw818 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part12
raw /dev/raw/raw819 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part12
raw /dev/raw/raw820 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part12
raw /dev/raw/raw821 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part12
raw /dev/raw/raw822 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part12
raw /dev/raw/raw823 /dev/disk/by-id/scsi-3600a0b80003694580000053f4852694c-part12
raw /dev/raw/raw824 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part12
raw /dev/raw/raw825 /dev/disk/by-id/scsi-3600a0b800036964400000aef485269b0-part12
raw /dev/raw/raw826 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part12
raw /dev/raw/raw827 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part12
raw /dev/raw/raw828 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part12
raw /dev/raw/raw829 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part12
raw /dev/raw/raw830 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part12
raw /dev/raw/raw831 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part12
raw /dev/raw/raw832 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part12
raw /dev/raw/raw833 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part12
raw /dev/raw/raw834 /dev/disk/by-id/scsi-3600a0b800034d5a800006bb48524108-part12
raw /dev/raw/raw835 /dev/disk/by-id/scsi-3600a0b800036fef0000084b48526d5c-part12

raw /dev/raw/raw836 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part12
raw /dev/raw/raw837 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part12
raw /dev/raw/raw838 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part12
raw /dev/raw/raw839 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part12
raw /dev/raw/raw840 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part12
raw /dev/raw/raw841 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part12
raw /dev/raw/raw842 /dev/disk/by-id/scsi-3600a0b800034d762000068448524232-part12
raw /dev/raw/raw843 /dev/disk/by-id/scsi-3600a0b80002458c6000003f48526584-part12
raw /dev/raw/raw844 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part12
raw /dev/raw/raw845 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part12
raw /dev/raw/raw846 /dev/disk/by-id/scsi-3600a0b8000245961000004004853a713-part12
raw /dev/raw/raw847 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part12
raw /dev/raw/raw848 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part12
raw /dev/raw/raw849 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part12
raw /dev/raw/raw850 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part12
raw /dev/raw/raw851 /dev/disk/by-id/scsi-3600a0b800034a9834000006d948526f37-part12
raw /dev/raw/raw852 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part12
raw /dev/raw/raw853 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part12
raw /dev/raw/raw854 /dev/disk/by-id/scsi-3600a0b80003ac5700000096b4852708d-part12
raw /dev/raw/raw855 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part12
raw /dev/raw/raw856 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a485262f2-part12
raw /dev/raw/raw857 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part12
raw /dev/raw/raw858 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part12
raw /dev/raw/raw859 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part12
raw /dev/raw/raw860 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part12
raw /dev/raw/raw861 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part12
raw /dev/raw/raw862 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part12
raw /dev/raw/raw863 /dev/disk/by-id/scsi-3600a0b80003a9a800000815485266e8-part12
raw /dev/raw/raw864 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part12
raw /dev/raw/raw865 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part12
raw /dev/raw/raw866 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part12
raw /dev/raw/raw867 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part12
raw /dev/raw/raw868 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part12
raw /dev/raw/raw869 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part12
raw /dev/raw/raw870 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part12

raw /dev/raw/raw871 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part12
raw /dev/raw/raw872 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part12
raw /dev/raw/raw873 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part12
raw /dev/raw/raw874 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part12
raw /dev/raw/raw875 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part12
raw /dev/raw/raw876 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part12
raw /dev/raw/raw877 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part12
raw /dev/raw/raw878 /dev/disk/by-id/scsi-3600a0b80003a98d0000062448526fad-part12
raw /dev/raw/raw879 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part12
raw /dev/raw/raw880 /dev/disk/by-id/scsi-3600a0b80003ac56b0000030648527075-part12

raw /dev/raw/raw881 /dev/disk/by-id/scsi-3600a0b800034d63d00000f3b48519090-part13
raw /dev/raw/raw882 /dev/disk/by-id/scsi-3600a0b800034999f0000112485190fc-part13
raw /dev/raw/raw883 /dev/disk/by-id/scsi-3600a0b80003498e5000018474851a164-part13
raw /dev/raw/raw884 /dev/disk/by-id/scsi-3600a0b800034d64c000011414851925c-part13
raw /dev/raw/raw885 /dev/disk/by-id/scsi-3600a0b800034d6290000113a4852394a-part13
raw /dev/raw/raw886 /dev/disk/by-id/scsi-3600a0b800034d6790000c4348519bb7-part13
raw /dev/raw/raw887 /dev/disk/by-id/scsi-3600a0b8000349916000002d48519ea7-part13
raw /dev/raw/raw888 /dev/disk/by-id/scsi-3600a0b800034d5db000015014851a14f-part13
raw /dev/raw/raw889 /dev/disk/by-id/scsi-3600a0b80003498e00000c6c4851a121-part13
raw /dev/raw/raw890 /dev/disk/by-id/scsi-3600a0b800034d5db00000c344851243f9-part13
raw /dev/raw/raw891 /dev/disk/by-id/scsi-3600a0b8000349998000009348519482-part13
raw /dev/raw/raw892 /dev/disk/by-id/scsi-3600a0b80003460e900000add485193d2-part13
raw /dev/raw/raw893 /dev/disk/by-id/scsi-3600a0b8000349932000009c948528f7a-part13
raw /dev/raw/raw894 /dev/disk/by-id/scsi-3600a0b800034d5ce00000a3e48524970-part13
raw /dev/raw/raw895 /dev/disk/by-id/scsi-3600a0b80003499350000121848523bca-part13
raw /dev/raw/raw896 /dev/disk/by-id/scsi-3600a0b800034999e00000c7c4851a159-part13
raw /dev/raw/raw897 /dev/disk/by-id/scsi-3600a0b80003460890000082048519cb3-part13
raw /dev/raw/raw898 /dev/disk/by-id/scsi-3600a0b8000349919000008e448529987-part13
raw /dev/raw/raw899 /dev/disk/by-id/scsi-3600a0b800034990d00000816485247f5-part13
raw /dev/raw/raw900 /dev/disk/by-id/scsi-3600a0b8000349918000011c648524411-part13
raw /dev/raw/raw901 /dev/disk/by-id/scsi-3600a0b80003698540000056c4852661c-part13
raw /dev/raw/raw902 /dev/disk/by-id/scsi-3600a0b800036991b000006b148526812-part13
raw /dev/raw/raw903 /dev/disk/by-id/scsi-3600a0b80003694850000053f4852694c-part13
raw /dev/raw/raw904 /dev/disk/by-id/scsi-3600a0b800034d6cd000007fc48523cf8-part13

raw /dev/raw/raw905 /dev/disk/by-id/scsi-3600a0b800036964400000aef485269b0-part13
raw /dev/raw/raw906 /dev/disk/by-id/scsi-3600a0b80003696900000056c4852729b-part13
raw /dev/raw/raw907 /dev/disk/by-id/scsi-3600a0b8000369864000005a44852768d-part13
raw /dev/raw/raw908 /dev/disk/by-id/scsi-3600a0b800036967f000004f84852736b-part13
raw /dev/raw/raw909 /dev/disk/by-id/scsi-3600a0b800034d6bd00000697485249a5-part13
raw /dev/raw/raw910 /dev/disk/by-id/scsi-3600a0b8000369e2d0000087e485275b5-part13
raw /dev/raw/raw911 /dev/disk/by-id/scsi-3600a0b80003460b70000077b48523fb0-part13
raw /dev/raw/raw912 /dev/disk/by-id/scsi-3600a0b800024589800000465485265bc-part13
raw /dev/raw/raw913 /dev/disk/by-id/scsi-3600a0b800034d6ce000005e748523e10-part13
raw /dev/raw/raw914 /dev/disk/by-id/scsi-3600a0b800034d5a800006bbb48524108-part13
raw /dev/raw/raw915 /dev/disk/by-id/scsi-3600a0b800036f6fd0000084b48526d5c-part13
raw /dev/raw/raw916 /dev/disk/by-id/scsi-3600a0b800034608d000010c948524c29-part13
raw /dev/raw/raw917 /dev/disk/by-id/scsi-3600a0b800024585100000469485271a3-part13
raw /dev/raw/raw918 /dev/disk/by-id/scsi-3600a0b800034d6bb0000052248524b27-part13
raw /dev/raw/raw919 /dev/disk/by-id/scsi-3600a0b800034d70b0000069a48524d0d-part13
raw /dev/raw/raw920 /dev/disk/by-id/scsi-3600a0b800034d6c50000079f48524b63-part13
raw /dev/raw/raw921 /dev/disk/by-id/scsi-3600a0b800024588c0000065148526746-part13
raw /dev/raw/raw922 /dev/disk/by-id/scsi-3600a0b800034d762000068448524232-part13
raw /dev/raw/raw923 /dev/disk/by-id/scsi-3600a0b80002458c6000003f48526584-part13
raw /dev/raw/raw924 /dev/disk/by-id/scsi-3600a0b800034d6a60000071348524106-part13
raw /dev/raw/raw925 /dev/disk/by-id/scsi-3600a0b8000245deb000008b148524fa4-part13
raw /dev/raw/raw926 /dev/disk/by-id/scsi-3600a0b8000245961000004d04853a713-part13
raw /dev/raw/raw927 /dev/disk/by-id/scsi-3600a0b800034d59d000006ad48524d37-part13
raw /dev/raw/raw928 /dev/disk/by-id/scsi-3600a0b800024589a0000042f485273e7-part13
raw /dev/raw/raw929 /dev/disk/by-id/scsi-3600a0b800034d58f0000061d48524df5-part13
raw /dev/raw/raw930 /dev/disk/by-id/scsi-3600a0b8000245d9600000690485260e1-part13
raw /dev/raw/raw931 /dev/disk/by-id/scsi-3600a0b80003a9834000006d948526f37-part13
raw /dev/raw/raw932 /dev/disk/by-id/scsi-3600a0b80003ac52d000006364852706d-part13
raw /dev/raw/raw933 /dev/disk/by-id/scsi-3600a0b80003ac613000007bd485270d3-part13
raw /dev/raw/raw934 /dev/disk/by-id/scsi-3600a0b80003ac5700000096b4852708d-part13
raw /dev/raw/raw935 /dev/disk/by-id/scsi-3600a0b800039c6f60000033148526e53-part13
raw /dev/raw/raw936 /dev/disk/by-id/scsi-3600a0b80003ac56c0000079a48526f2f-part13
raw /dev/raw/raw937 /dev/disk/by-id/scsi-3600a0b80003ac5290000070248526452-part13
raw /dev/raw/raw938 /dev/disk/by-id/scsi-3600a0b80003ac5ef000008cf48526454-part13
raw /dev/raw/raw939 /dev/disk/by-id/scsi-3600a0b80003ac57900000a8c485263d6-part13

```
raw /dev/raw/raw940 /dev/disk/by-id/scsi-3600a0b800039caf8000003174852648e-part13
raw /dev/raw/raw941 /dev/disk/by-id/scsi-3600a0b80003ac55b000009f048526640-part13
raw /dev/raw/raw942 /dev/disk/by-id/scsi-3600a0b80003ac61c000006cf4852657a-part13
raw /dev/raw/raw943 /dev/disk/by-id/scsi-3600a0b80003a9a8000000815485266e8-part13
raw /dev/raw/raw944 /dev/disk/by-id/scsi-3600a0b80003ac5a10000081a4852663c-part13
raw /dev/raw/raw945 /dev/disk/by-id/scsi-3600a0b80003ac506000002f2485266bc-part13
raw /dev/raw/raw946 /dev/disk/by-id/scsi-3600a0b80003ac5640000091148526f73-part13
raw /dev/raw/raw947 /dev/disk/by-id/scsi-3600a0b80003ac4a20000062648526f2f-part13
raw /dev/raw/raw948 /dev/disk/by-id/scsi-3600a0b80003a9a350000077c48526fcd-part13
raw /dev/raw/raw949 /dev/disk/by-id/scsi-3600a0b80003ac64b0000079448527065-part13
raw /dev/raw/raw950 /dev/disk/by-id/scsi-3600a0b80003a991e0000030048526e9f-part13
raw /dev/raw/raw951 /dev/disk/by-id/scsi-3600a0b80003ac625000008434852672e-part13
raw /dev/raw/raw952 /dev/disk/by-id/scsi-3600a0b80003ac648000008e1485265ee-part13
raw /dev/raw/raw953 /dev/disk/by-id/scsi-3600a0b80003ac60b0000070048526826-part13
raw /dev/raw/raw954 /dev/disk/by-id/scsi-3600a0b80003a998600000721485268b4-part13
raw /dev/raw/raw955 /dev/disk/by-id/scsi-3600a0b80003ac4e30000030248526a56-part13
raw /dev/raw/raw956 /dev/disk/by-id/scsi-3600a0b80003ac5760000075148526ee9-part13
raw /dev/raw/raw957 /dev/disk/by-id/scsi-3600a0b80003ac67d000007d848526dcb-part13
raw /dev/raw/raw958 /dev/disk/by-id/scsi-3600a0b80003a98d00000062448526fad-part13
raw /dev/raw/raw959 /dev/disk/by-id/scsi-3600a0b80003a9a2f0000066e48527035-part13
raw /dev/raw/raw960 /dev/disk/by-id/scsi-3600a0b80003ac5b60000030648527075-part13
```

sleep 10

chmod 777 /dev/raw/raw*

version.txt

Linux itcopus83.austin.ibm.com 2.6.18-92.el5 #1 SMP Tue Apr 29 13:16:15 EDT 2008 x86_64 x86_64 x86_64 GNU/Linux

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 18001
AND 19200);
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 19201
AND 20400);
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 20401
AND 21600);
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 21601
AND 22800);
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
1200);
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 22801
AND 24000);
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 24001
AND 25200);
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 25201
AND 26400);

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 26401
AND 27600);
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 27601
AND 28800);
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 28801
AND 30000);
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 30001
AND 31200);
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 31201
AND 32400);
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 32401
AND 33600);
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 33601
AND 34800);
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 1201 AND
2400);
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 34801
AND 36000);
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 36001
AND 37200);
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 BETWEEN 37201
AND 38400);
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER33.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER33 OFF;
ALTER TABLE CUSTOMER33 DROP CONSTRAINT
CUSTOMER33CKC;
ALTER TABLE CUSTOMER33 ADD CONSTRAINT
CUSTOMER33CKC CHECK (C_W_ID BETWEEN 38401
AND 39600);
SET INTEGRITY FOR CUSTOMER33 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER34.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER34 OFF;
ALTER TABLE CUSTOMER34 DROP CONSTRAINT
CUSTOMER34CKC;
ALTER TABLE CUSTOMER34 ADD CONSTRAINT
CUSTOMER34CKC CHECK (C_W_ID BETWEEN 39601
AND 40800);
SET INTEGRITY FOR CUSTOMER34 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER35.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER35 OFF;
ALTER TABLE CUSTOMER35 DROP CONSTRAINT
CUSTOMER35CKC;
ALTER TABLE CUSTOMER35 ADD CONSTRAINT
CUSTOMER35CKC CHECK (C_W_ID BETWEEN 40801
AND 42000);
SET INTEGRITY FOR CUSTOMER35 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER36.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER36 OFF;
ALTER TABLE CUSTOMER36 DROP CONSTRAINT
CUSTOMER36CKC;
ALTER TABLE CUSTOMER36 ADD CONSTRAINT
CUSTOMER36CKC CHECK (C_W_ID BETWEEN 42001
AND 43200);
```

SET INTEGRITY FOR CUSTOMER36 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER37.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER37 OFF;
ALTER TABLE CUSTOMER37 DROP CONSTRAINT CUSTOMER37CKC;
ALTER TABLE CUSTOMER37 ADD CONSTRAINT CUSTOMER37CKC CHECK (C_W_ID BETWEEN 43201 AND 44400);
SET INTEGRITY FOR CUSTOMER37 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER38.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER38 OFF;
ALTER TABLE CUSTOMER38 DROP CONSTRAINT CUSTOMER38CKC;
ALTER TABLE CUSTOMER38 ADD CONSTRAINT CUSTOMER38CKC CHECK (C_W_ID BETWEEN 44401 AND 45600);
SET INTEGRITY FOR CUSTOMER38 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER39.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER39 OFF;
ALTER TABLE CUSTOMER39 DROP CONSTRAINT CUSTOMER39CKC;
ALTER TABLE CUSTOMER39 ADD CONSTRAINT CUSTOMER39CKC CHECK (C_W_ID BETWEEN 45601 AND 46800);
SET INTEGRITY FOR CUSTOMER39 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 2401 AND 3600);
SET INTEGRITY FOR CUSTOMER3 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER40.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER40 OFF;
ALTER TABLE CUSTOMER40 DROP CONSTRAINT CUSTOMER40CKC;
ALTER TABLE CUSTOMER40 ADD CONSTRAINT CUSTOMER40CKC CHECK (C_W_ID BETWEEN 46801 AND 48000);
SET INTEGRITY FOR CUSTOMER40 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER41.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER41 OFF;
ALTER TABLE CUSTOMER41 DROP CONSTRAINT CUSTOMER41CKC;
ALTER TABLE CUSTOMER41 ADD CONSTRAINT CUSTOMER41CKC CHECK (C_W_ID BETWEEN 48001 AND 49200);
SET INTEGRITY FOR CUSTOMER41 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER42.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER42 OFF;
ALTER TABLE CUSTOMER42 DROP CONSTRAINT CUSTOMER42CKC;
ALTER TABLE CUSTOMER42 ADD CONSTRAINT CUSTOMER42CKC CHECK (C_W_ID BETWEEN 49201 AND 50400);
SET INTEGRITY FOR CUSTOMER42 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER43.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER43 OFF;
ALTER TABLE CUSTOMER43 DROP CONSTRAINT CUSTOMER43CKC;
ALTER TABLE CUSTOMER43 ADD CONSTRAINT CUSTOMER43CKC CHECK (C_W_ID BETWEEN 50401 AND 51600);
SET INTEGRITY FOR CUSTOMER43 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER44.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER44 OFF;
ALTER TABLE CUSTOMER44 DROP CONSTRAINT CUSTOMER44CKC;
ALTER TABLE CUSTOMER44 ADD CONSTRAINT CUSTOMER44CKC CHECK (C_W_ID BETWEEN 51601 AND 52800);
SET INTEGRITY FOR CUSTOMER44 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER45.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER45 OFF;
ALTER TABLE CUSTOMER45 DROP CONSTRAINT CUSTOMER45CKC;
ALTER TABLE CUSTOMER45 ADD CONSTRAINT CUSTOMER45CKC CHECK (C_W_ID BETWEEN 52801 AND 54000);
SET INTEGRITY FOR CUSTOMER45 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER46.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER46 OFF;
ALTER TABLE CUSTOMER46 DROP CONSTRAINT CUSTOMER46CKC;
ALTER TABLE CUSTOMER46 ADD CONSTRAINT CUSTOMER46CKC CHECK (C_W_ID BETWEEN 54001 AND 55200);
SET INTEGRITY FOR CUSTOMER46 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST CUSTOMER47.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER47 OFF;
ALTER TABLE CUSTOMER47 DROP CONSTRAINT
CUSTOMER47CKC;
ALTER TABLE CUSTOMER47 ADD CONSTRAINT
CUSTOMER47CKC CHECK (C_W_ID BETWEEN 55201
AND 56400);
SET INTEGRITY FOR CUSTOMER47 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER48.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER48 OFF;
ALTER TABLE CUSTOMER48 DROP CONSTRAINT
CUSTOMER48CKC;
ALTER TABLE CUSTOMER48 ADD CONSTRAINT
CUSTOMER48CKC CHECK (C_W_ID BETWEEN 56401
AND 57600);
SET INTEGRITY FOR CUSTOMER48 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER49.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER49 OFF;
ALTER TABLE CUSTOMER49 DROP CONSTRAINT
CUSTOMER49CKC;
ALTER TABLE CUSTOMER49 ADD CONSTRAINT
CUSTOMER49CKC CHECK (C_W_ID BETWEEN 57601
AND 58800);
SET INTEGRITY FOR CUSTOMER49 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 3601 AND
4800);

SET INTEGRITY FOR CUSTOMER4 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER50.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER50 OFF;
ALTER TABLE CUSTOMER50 DROP CONSTRAINT
CUSTOMER50CKC;
ALTER TABLE CUSTOMER50 ADD CONSTRAINT
CUSTOMER50CKC CHECK (C_W_ID BETWEEN 58801
AND 60000);
SET INTEGRITY FOR CUSTOMER50 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER51.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER51 OFF;
ALTER TABLE CUSTOMER51 DROP CONSTRAINT
CUSTOMER51CKC;
ALTER TABLE CUSTOMER51 ADD CONSTRAINT
CUSTOMER51CKC CHECK (C_W_ID BETWEEN 60001
AND 61200);
SET INTEGRITY FOR CUSTOMER51 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER52.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER52 OFF;
ALTER TABLE CUSTOMER52 DROP CONSTRAINT
CUSTOMER52CKC;
ALTER TABLE CUSTOMER52 ADD CONSTRAINT
CUSTOMER52CKC CHECK (C_W_ID BETWEEN 61201
AND 62400);
SET INTEGRITY FOR CUSTOMER52 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER53.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER53 OFF;

ALTER TABLE CUSTOMER53 DROP CONSTRAINT
CUSTOMER53CKC;
ALTER TABLE CUSTOMER53 ADD CONSTRAINT
CUSTOMER53CKC CHECK (C_W_ID BETWEEN 62401
AND 63600);
SET INTEGRITY FOR CUSTOMER53 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER54.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER54 OFF;
ALTER TABLE CUSTOMER54 DROP CONSTRAINT
CUSTOMER54CKC;
ALTER TABLE CUSTOMER54 ADD CONSTRAINT
CUSTOMER54CKC CHECK (C_W_ID BETWEEN 63601
AND 64800);
SET INTEGRITY FOR CUSTOMER54 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER55.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER55 OFF;
ALTER TABLE CUSTOMER55 DROP CONSTRAINT
CUSTOMER55CKC;
ALTER TABLE CUSTOMER55 ADD CONSTRAINT
CUSTOMER55CKC CHECK (C_W_ID BETWEEN 64801
AND 66000);
SET INTEGRITY FOR CUSTOMER55 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER56.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER56 OFF;
ALTER TABLE CUSTOMER56 DROP CONSTRAINT
CUSTOMER56CKC;
ALTER TABLE CUSTOMER56 ADD CONSTRAINT
CUSTOMER56CKC CHECK (C_W_ID BETWEEN 66001
AND 67200);
SET INTEGRITY FOR CUSTOMER56 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER57.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER57 OFF;
ALTER TABLE CUSTOMER57 DROP CONSTRAINT
CUSTOMER57CKC;
ALTER TABLE CUSTOMER57 ADD CONSTRAINT
CUSTOMER57CKC CHECK (C_W_ID BETWEEN 67201
AND 68400);
SET INTEGRITY FOR CUSTOMER57 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER58.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER58 OFF;
ALTER TABLE CUSTOMER58 DROP CONSTRAINT
CUSTOMER58CKC;
ALTER TABLE CUSTOMER58 ADD CONSTRAINT
CUSTOMER58CKC CHECK (C_W_ID BETWEEN 68401
AND 69600);
SET INTEGRITY FOR CUSTOMER58 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER59.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER59 OFF;
ALTER TABLE CUSTOMER59 DROP CONSTRAINT
CUSTOMER59CKC;
ALTER TABLE CUSTOMER59 ADD CONSTRAINT
CUSTOMER59CKC CHECK (C_W_ID BETWEEN 69601
AND 70800);
SET INTEGRITY FOR CUSTOMER59 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 4801 AND
6000);
SET INTEGRITY FOR CUSTOMER5 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER60.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER60 OFF;
ALTER TABLE CUSTOMER60 DROP CONSTRAINT
CUSTOMER60CKC;
ALTER TABLE CUSTOMER60 ADD CONSTRAINT
CUSTOMER60CKC CHECK (C_W_ID BETWEEN 70801
AND 72000);
SET INTEGRITY FOR CUSTOMER60 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER61.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER61 OFF;
ALTER TABLE CUSTOMER61 DROP CONSTRAINT
CUSTOMER61CKC;
ALTER TABLE CUSTOMER61 ADD CONSTRAINT
CUSTOMER61CKC CHECK (C_W_ID BETWEEN 72001
AND 73200);
SET INTEGRITY FOR CUSTOMER61 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER62.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER62 OFF;
ALTER TABLE CUSTOMER62 DROP CONSTRAINT
CUSTOMER62CKC;
ALTER TABLE CUSTOMER62 ADD CONSTRAINT
CUSTOMER62CKC CHECK (C_W_ID BETWEEN 73201
AND 74400);
SET INTEGRITY FOR CUSTOMER62 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER63.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER63 OFF;
ALTER TABLE CUSTOMER63 DROP CONSTRAINT
CUSTOMER63CKC;
ALTER TABLE CUSTOMER63 ADD CONSTRAINT
CUSTOMER63CKC CHECK (C_W_ID BETWEEN 74401
AND 75600);

SET INTEGRITY FOR CUSTOMER63 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER64.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER64 OFF;
ALTER TABLE CUSTOMER64 DROP CONSTRAINT
CUSTOMER64CKC;
ALTER TABLE CUSTOMER64 ADD CONSTRAINT
CUSTOMER64CKC CHECK (C_W_ID BETWEEN 75601
AND 76800);
SET INTEGRITY FOR CUSTOMER64 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER65.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER65 OFF;
ALTER TABLE CUSTOMER65 DROP CONSTRAINT
CUSTOMER65CKC;
ALTER TABLE CUSTOMER65 ADD CONSTRAINT
CUSTOMER65CKC CHECK (C_W_ID BETWEEN 76801
AND 78000);
SET INTEGRITY FOR CUSTOMER65 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER66.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER66 OFF;
ALTER TABLE CUSTOMER66 DROP CONSTRAINT
CUSTOMER66CKC;
ALTER TABLE CUSTOMER66 ADD CONSTRAINT
CUSTOMER66CKC CHECK (C_W_ID BETWEEN 78001
AND 79200);
SET INTEGRITY FOR CUSTOMER66 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST CUSTOMER67.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER67 OFF;

```
ALTER TABLE CUSTOMER67 DROP CONSTRAINT
CUSTOMER67CKC;
ALTER TABLE CUSTOMER67 ADD CONSTRAINT
CUSTOMER67CKC CHECK (C_W_ID BETWEEN 79201
AND 80400);
SET INTEGRITY FOR CUSTOMER67 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER68.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER68 OFF;
ALTER TABLE CUSTOMER68 DROP CONSTRAINT
CUSTOMER68CKC;
ALTER TABLE CUSTOMER68 ADD CONSTRAINT
CUSTOMER68CKC CHECK (C_W_ID BETWEEN 80401
AND 81600);
SET INTEGRITY FOR CUSTOMER68 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER69.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER69 OFF;
ALTER TABLE CUSTOMER69 DROP CONSTRAINT
CUSTOMER69CKC;
ALTER TABLE CUSTOMER69 ADD CONSTRAINT
CUSTOMER69CKC CHECK (C_W_ID BETWEEN 81601
AND 82800);
SET INTEGRITY FOR CUSTOMER69 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 6001 AND
7200);
SET INTEGRITY FOR CUSTOMER6 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER70.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER70 OFF;
ALTER TABLE CUSTOMER70 DROP CONSTRAINT
CUSTOMER70CKC;
ALTER TABLE CUSTOMER70 ADD CONSTRAINT
CUSTOMER70CKC CHECK (C_W_ID BETWEEN 82801
AND 84000);
SET INTEGRITY FOR CUSTOMER70 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER71.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER71 OFF;
ALTER TABLE CUSTOMER71 DROP CONSTRAINT
CUSTOMER71CKC;
ALTER TABLE CUSTOMER71 ADD CONSTRAINT
CUSTOMER71CKC CHECK (C_W_ID BETWEEN 84001
AND 85200);
SET INTEGRITY FOR CUSTOMER71 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER72.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER72 OFF;
ALTER TABLE CUSTOMER72 DROP CONSTRAINT
CUSTOMER72CKC;
ALTER TABLE CUSTOMER72 ADD CONSTRAINT
CUSTOMER72CKC CHECK (C_W_ID BETWEEN 85201
AND 86400);
SET INTEGRITY FOR CUSTOMER72 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER73.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER73 OFF;
ALTER TABLE CUSTOMER73 DROP CONSTRAINT
CUSTOMER73CKC;
ALTER TABLE CUSTOMER73 ADD CONSTRAINT
CUSTOMER73CKC CHECK (C_W_ID BETWEEN 86401
AND 87600);
SET INTEGRITY FOR CUSTOMER73 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER74.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER74 OFF;
ALTER TABLE CUSTOMER74 DROP CONSTRAINT
CUSTOMER74CKC;
ALTER TABLE CUSTOMER74 ADD CONSTRAINT
CUSTOMER74CKC CHECK (C_W_ID BETWEEN 87601
AND 88800);
SET INTEGRITY FOR CUSTOMER74 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER75.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER75 OFF;
ALTER TABLE CUSTOMER75 DROP CONSTRAINT
CUSTOMER75CKC;
ALTER TABLE CUSTOMER75 ADD CONSTRAINT
CUSTOMER75CKC CHECK (C_W_ID BETWEEN 88801
AND 90000);
SET INTEGRITY FOR CUSTOMER75 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER76.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER76 OFF;
ALTER TABLE CUSTOMER76 DROP CONSTRAINT
CUSTOMER76CKC;
ALTER TABLE CUSTOMER76 ADD CONSTRAINT
CUSTOMER76CKC CHECK (C_W_ID BETWEEN 90001
AND 91200);
SET INTEGRITY FOR CUSTOMER76 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER77.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER77 OFF;
ALTER TABLE CUSTOMER77 DROP CONSTRAINT
CUSTOMER77CKC;
ALTER TABLE CUSTOMER77 ADD CONSTRAINT
CUSTOMER77CKC CHECK (C_W_ID BETWEEN 91201
AND 92400);
```

```
SET INTEGRITY FOR CUSTOMER77 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER78.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER78 OFF;
ALTER TABLE CUSTOMER78 DROP CONSTRAINT
CUSTOMER78CKC;
ALTER TABLE CUSTOMER78 ADD CONSTRAINT
CUSTOMER78CKC CHECK (C_W_ID BETWEEN 92401
AND 93600);
SET INTEGRITY FOR CUSTOMER78 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER79.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER79 OFF;
ALTER TABLE CUSTOMER79 DROP CONSTRAINT
CUSTOMER79CKC;
ALTER TABLE CUSTOMER79 ADD CONSTRAINT
CUSTOMER79CKC CHECK (C_W_ID BETWEEN 93601
AND 94800);
SET INTEGRITY FOR CUSTOMER79 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 7201 AND
8400);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST CUSTOMER80.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER80 OFF;
```

```
ALTER TABLE CUSTOMER80 DROP CONSTRAINT
CUSTOMER80CKC;
ALTER TABLE CUSTOMER80 ADD CONSTRAINT
CUSTOMER80CKC CHECK (C_W_ID >= 94801);
SET INTEGRITY FOR CUSTOMER80 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 8401 AND
9600);
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 9601 AND
10800);
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 10801 AND
12000);
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 12001 AND
13200);
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 13201 AND
14400);
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 14401 AND
15600);
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 15601 AND
16800);
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 16801 AND
18000);
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 18001 AND
19200);
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 19201 AND
20400);
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 20401 AND
21600);

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 21601 AND
22800);
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 1200);
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 22801 AND
24000);
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 24001 AND
25200);
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 25201 AND
26400);
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 26401 AND
27600);
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 27601 AND
28800);
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 28801 AND
30000);
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 30001 AND
31200);
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 31201 AND
32400);
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 32401 AND
33600);
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 33601 AND
34800);
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 1201 AND
2400);
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 34801 AND
36000);
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 36001 AND
37200);

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 BETWEEN 37201 AND
38400);
SET INTEGRITY FOR DISTRICT32 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT33.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT33 OFF;
ALTER TABLE DISTRICT33 DROP CONSTRAINT
DISTRICT33CKC;
ALTER TABLE DISTRICT33 ADD CONSTRAINT
DISTRICT33CKC CHECK (D_W_ID BETWEEN 38401 AND
39600);
SET INTEGRITY FOR DISTRICT33 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT34.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT34 OFF;
ALTER TABLE DISTRICT34 DROP CONSTRAINT
DISTRICT34CKC;
ALTER TABLE DISTRICT34 ADD CONSTRAINT
DISTRICT34CKC CHECK (D_W_ID BETWEEN 39601 AND
40800);
SET INTEGRITY FOR DISTRICT34 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT35.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT35 OFF;

```
ALTER TABLE DISTRICT35 DROP CONSTRAINT
DISTRICT35CKC;
ALTER TABLE DISTRICT35 ADD CONSTRAINT
DISTRICT35CKC CHECK (D_W_ID BETWEEN 40801 AND
42000);
SET INTEGRITY FOR DISTRICT35 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT36.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT36 OFF;
ALTER TABLE DISTRICT36 DROP CONSTRAINT
DISTRICT36CKC;
ALTER TABLE DISTRICT36 ADD CONSTRAINT
DISTRICT36CKC CHECK (D_W_ID BETWEEN 42001 AND
43200);
SET INTEGRITY FOR DISTRICT36 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT37.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT37 OFF;
ALTER TABLE DISTRICT37 DROP CONSTRAINT
DISTRICT37CKC;
ALTER TABLE DISTRICT37 ADD CONSTRAINT
DISTRICT37CKC CHECK (D_W_ID BETWEEN 43201 AND
44400);
SET INTEGRITY FOR DISTRICT37 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT38.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT38 OFF;
ALTER TABLE DISTRICT38 DROP CONSTRAINT
DISTRICT38CKC;
ALTER TABLE DISTRICT38 ADD CONSTRAINT
DISTRICT38CKC CHECK (D_W_ID BETWEEN 44401 AND
45600);
SET INTEGRITY FOR DISTRICT38 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT39.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT39 OFF;
ALTER TABLE DISTRICT39 DROP CONSTRAINT
DISTRICT39CKC;
ALTER TABLE DISTRICT39 ADD CONSTRAINT
DISTRICT39CKC CHECK (D_W_ID BETWEEN 45601 AND
46800);
SET INTEGRITY FOR DISTRICT39 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 2401 AND
3600);
SET INTEGRITY FOR DISTRICT3 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT40.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT40 OFF;
ALTER TABLE DISTRICT40 DROP CONSTRAINT
DISTRICT40CKC;
ALTER TABLE DISTRICT40 ADD CONSTRAINT
DISTRICT40CKC CHECK (D_W_ID BETWEEN 46801 AND
48000);
SET INTEGRITY FOR DISTRICT40 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT41.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT41 OFF;
ALTER TABLE DISTRICT41 DROP CONSTRAINT
DISTRICT41CKC;
ALTER TABLE DISTRICT41 ADD CONSTRAINT
DISTRICT41CKC CHECK (D_W_ID BETWEEN 48001 AND
49200);
SET INTEGRITY FOR DISTRICT41 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT42.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT42 OFF;
ALTER TABLE DISTRICT42 DROP CONSTRAINT
DISTRICT42CKC;
ALTER TABLE DISTRICT42 ADD CONSTRAINT
DISTRICT42CKC CHECK (D_W_ID BETWEEN 49201 AND
50400);
SET INTEGRITY FOR DISTRICT42 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT43.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT43 OFF;
ALTER TABLE DISTRICT43 DROP CONSTRAINT
DISTRICT43CKC;
ALTER TABLE DISTRICT43 ADD CONSTRAINT
DISTRICT43CKC CHECK (D_W_ID BETWEEN 50401 AND
51600);
SET INTEGRITY FOR DISTRICT43 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT44.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT44 OFF;
ALTER TABLE DISTRICT44 DROP CONSTRAINT
DISTRICT44CKC;
ALTER TABLE DISTRICT44 ADD CONSTRAINT
DISTRICT44CKC CHECK (D_W_ID BETWEEN 51601 AND
52800);
SET INTEGRITY FOR DISTRICT44 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT45.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT45 OFF;
ALTER TABLE DISTRICT45 DROP CONSTRAINT
DISTRICT45CKC;
ALTER TABLE DISTRICT45 ADD CONSTRAINT
DISTRICT45CKC CHECK (D_W_ID BETWEEN 52801 AND
54000);
```

SET INTEGRITY FOR DISTRICT45 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT46.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT46 OFF;
ALTER TABLE DISTRICT46 DROP CONSTRAINT
DISTRICT46CKC;
ALTER TABLE DISTRICT46 ADD CONSTRAINT
DISTRICT46CKC CHECK (D_W_ID BETWEEN 54001 AND
55200);
SET INTEGRITY FOR DISTRICT46 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT47.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT47 OFF;
ALTER TABLE DISTRICT47 DROP CONSTRAINT
DISTRICT47CKC;
ALTER TABLE DISTRICT47 ADD CONSTRAINT
DISTRICT47CKC CHECK (D_W_ID BETWEEN 55201 AND
56400);
SET INTEGRITY FOR DISTRICT47 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT48.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT48 OFF;
ALTER TABLE DISTRICT48 DROP CONSTRAINT
DISTRICT48CKC;
ALTER TABLE DISTRICT48 ADD CONSTRAINT
DISTRICT48CKC CHECK (D_W_ID BETWEEN 56401 AND
57600);
SET INTEGRITY FOR DISTRICT48 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT49.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT49 OFF;

ALTER TABLE DISTRICT49 DROP CONSTRAINT
DISTRICT49CKC;
ALTER TABLE DISTRICT49 ADD CONSTRAINT
DISTRICT49CKC CHECK (D_W_ID BETWEEN 57601 AND
58800);
SET INTEGRITY FOR DISTRICT49 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 3601 AND
4800);
SET INTEGRITY FOR DISTRICT4 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT50.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT50 OFF;
ALTER TABLE DISTRICT50 DROP CONSTRAINT
DISTRICT50CKC;
ALTER TABLE DISTRICT50 ADD CONSTRAINT
DISTRICT50CKC CHECK (D_W_ID BETWEEN 58801 AND
60000);
SET INTEGRITY FOR DISTRICT50 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT51.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT51 OFF;
ALTER TABLE DISTRICT51 DROP CONSTRAINT
DISTRICT51CKC;
ALTER TABLE DISTRICT51 ADD CONSTRAINT
DISTRICT51CKC CHECK (D_W_ID BETWEEN 60001 AND
61200);
SET INTEGRITY FOR DISTRICT51 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT52.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT52 OFF;
ALTER TABLE DISTRICT52 DROP CONSTRAINT
DISTRICT52CKC;
ALTER TABLE DISTRICT52 ADD CONSTRAINT
DISTRICT52CKC CHECK (D_W_ID BETWEEN 61201 AND
62400);
SET INTEGRITY FOR DISTRICT52 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT53.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT53 OFF;
ALTER TABLE DISTRICT53 DROP CONSTRAINT
DISTRICT53CKC;
ALTER TABLE DISTRICT53 ADD CONSTRAINT
DISTRICT53CKC CHECK (D_W_ID BETWEEN 62401 AND
63600);
SET INTEGRITY FOR DISTRICT53 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT54.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT54 OFF;
ALTER TABLE DISTRICT54 DROP CONSTRAINT
DISTRICT54CKC;
ALTER TABLE DISTRICT54 ADD CONSTRAINT
DISTRICT54CKC CHECK (D_W_ID BETWEEN 63601 AND
64800);
SET INTEGRITY FOR DISTRICT54 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST DISTRICT55.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT55 OFF;
ALTER TABLE DISTRICT55 DROP CONSTRAINT
DISTRICT55CKC;
ALTER TABLE DISTRICT55 ADD CONSTRAINT
DISTRICT55CKC CHECK (D_W_ID BETWEEN 64801 AND
66000);
SET INTEGRITY FOR DISTRICT55 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT56.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT56 OFF;
ALTER TABLE DISTRICT56 DROP CONSTRAINT
DISTRICT56CKC;
ALTER TABLE DISTRICT56 ADD CONSTRAINT
DISTRICT56CKC CHECK (D_W_ID BETWEEN 66001 AND
67200);
SET INTEGRITY FOR DISTRICT56 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT57.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT57 OFF;
ALTER TABLE DISTRICT57 DROP CONSTRAINT
DISTRICT57CKC;
ALTER TABLE DISTRICT57 ADD CONSTRAINT
DISTRICT57CKC CHECK (D_W_ID BETWEEN 67201 AND
68400);
SET INTEGRITY FOR DISTRICT57 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT58.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT58 OFF;
ALTER TABLE DISTRICT58 DROP CONSTRAINT
DISTRICT58CKC;
ALTER TABLE DISTRICT58 ADD CONSTRAINT
DISTRICT58CKC CHECK (D_W_ID BETWEEN 68401 AND
69600);
SET INTEGRITY FOR DISTRICT58 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT59.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT59 OFF;
ALTER TABLE DISTRICT59 DROP CONSTRAINT
DISTRICT59CKC;
ALTER TABLE DISTRICT59 ADD CONSTRAINT
DISTRICT59CKC CHECK (D_W_ID BETWEEN 69601 AND
70800);

SET INTEGRITY FOR DISTRICT59 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 4801 AND
6000);
SET INTEGRITY FOR DISTRICT5 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT60.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT60 OFF;
ALTER TABLE DISTRICT60 DROP CONSTRAINT
DISTRICT60CKC;
ALTER TABLE DISTRICT60 ADD CONSTRAINT
DISTRICT60CKC CHECK (D_W_ID BETWEEN 70801 AND
72000);
SET INTEGRITY FOR DISTRICT60 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT61.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT61 OFF;
ALTER TABLE DISTRICT61 DROP CONSTRAINT
DISTRICT61CKC;
ALTER TABLE DISTRICT61 ADD CONSTRAINT
DISTRICT61CKC CHECK (D_W_ID BETWEEN 72001 AND
73200);
SET INTEGRITY FOR DISTRICT61 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT62.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT62 OFF;

ALTER TABLE DISTRICT62 DROP CONSTRAINT
DISTRICT62CKC;
ALTER TABLE DISTRICT62 ADD CONSTRAINT
DISTRICT62CKC CHECK (D_W_ID BETWEEN 73201 AND
74400);
SET INTEGRITY FOR DISTRICT62 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT63.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT63 OFF;
ALTER TABLE DISTRICT63 DROP CONSTRAINT
DISTRICT63CKC;
ALTER TABLE DISTRICT63 ADD CONSTRAINT
DISTRICT63CKC CHECK (D_W_ID BETWEEN 74401 AND
75600);
SET INTEGRITY FOR DISTRICT63 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT64.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT64 OFF;
ALTER TABLE DISTRICT64 DROP CONSTRAINT
DISTRICT64CKC;
ALTER TABLE DISTRICT64 ADD CONSTRAINT
DISTRICT64CKC CHECK (D_W_ID BETWEEN 75601 AND
76800);
SET INTEGRITY FOR DISTRICT64 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT65.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT65 OFF;
ALTER TABLE DISTRICT65 DROP CONSTRAINT
DISTRICT65CKC;
ALTER TABLE DISTRICT65 ADD CONSTRAINT
DISTRICT65CKC CHECK (D_W_ID BETWEEN 76801 AND
78000);
SET INTEGRITY FOR DISTRICT65 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT66.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT66 OFF;
ALTER TABLE DISTRICT66 DROP CONSTRAINT
DISTRICT66CKC;
ALTER TABLE DISTRICT66 ADD CONSTRAINT
DISTRICT66CKC CHECK (D_W_ID BETWEEN 78001 AND
79200);
SET INTEGRITY FOR DISTRICT66 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT67.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT67 OFF;
ALTER TABLE DISTRICT67 DROP CONSTRAINT
DISTRICT67CKC;
ALTER TABLE DISTRICT67 ADD CONSTRAINT
DISTRICT67CKC CHECK (D_W_ID BETWEEN 79201 AND
80400);
SET INTEGRITY FOR DISTRICT67 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT68.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT68 OFF;
ALTER TABLE DISTRICT68 DROP CONSTRAINT
DISTRICT68CKC;
ALTER TABLE DISTRICT68 ADD CONSTRAINT
DISTRICT68CKC CHECK (D_W_ID BETWEEN 80401 AND
81600);
SET INTEGRITY FOR DISTRICT68 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT69.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT69 OFF;
ALTER TABLE DISTRICT69 DROP CONSTRAINT
DISTRICT69CKC;
ALTER TABLE DISTRICT69 ADD CONSTRAINT
DISTRICT69CKC CHECK (D_W_ID BETWEEN 81601 AND
82800);
SET INTEGRITY FOR DISTRICT69 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 6001 AND
7200);
SET INTEGRITY FOR DISTRICT6 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT70.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT70 OFF;
ALTER TABLE DISTRICT70 DROP CONSTRAINT
DISTRICT70CKC;
ALTER TABLE DISTRICT70 ADD CONSTRAINT
DISTRICT70CKC CHECK (D_W_ID BETWEEN 82801 AND
84000);
SET INTEGRITY FOR DISTRICT70 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT71.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT71 OFF;
ALTER TABLE DISTRICT71 DROP CONSTRAINT
DISTRICT71CKC;
ALTER TABLE DISTRICT71 ADD CONSTRAINT
DISTRICT71CKC CHECK (D_W_ID BETWEEN 84001 AND
85200);
SET INTEGRITY FOR DISTRICT71 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT72.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT72 OFF;
ALTER TABLE DISTRICT72 DROP CONSTRAINT
DISTRICT72CKC;
ALTER TABLE DISTRICT72 ADD CONSTRAINT
DISTRICT72CKC CHECK (D_W_ID BETWEEN 85201 AND
86400);

SET INTEGRITY FOR DISTRICT72 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT73.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT73 OFF;
ALTER TABLE DISTRICT73 DROP CONSTRAINT
DISTRICT73CKC;
ALTER TABLE DISTRICT73 ADD CONSTRAINT
DISTRICT73CKC CHECK (D_W_ID BETWEEN 86401 AND
87600);
SET INTEGRITY FOR DISTRICT73 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT74.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT74 OFF;
ALTER TABLE DISTRICT74 DROP CONSTRAINT
DISTRICT74CKC;
ALTER TABLE DISTRICT74 ADD CONSTRAINT
DISTRICT74CKC CHECK (D_W_ID BETWEEN 87601 AND
88800);
SET INTEGRITY FOR DISTRICT74 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT75.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT75 OFF;
ALTER TABLE DISTRICT75 DROP CONSTRAINT
DISTRICT75CKC;
ALTER TABLE DISTRICT75 ADD CONSTRAINT
DISTRICT75CKC CHECK (D_W_ID BETWEEN 88801 AND
90000);
SET INTEGRITY FOR DISTRICT75 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_DISTRICT76.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT76 OFF;

```
ALTER TABLE DISTRICT76 DROP CONSTRAINT
DISTRICT76CKC;
ALTER TABLE DISTRICT76 ADD CONSTRAINT
DISTRICT76CKC CHECK (D_W_ID BETWEEN 90001 AND
91200);
SET INTEGRITY FOR DISTRICT76 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT77.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT77 OFF;
ALTER TABLE DISTRICT77 DROP CONSTRAINT
DISTRICT77CKC;
ALTER TABLE DISTRICT77 ADD CONSTRAINT
DISTRICT77CKC CHECK (D_W_ID BETWEEN 91201 AND
92400);
SET INTEGRITY FOR DISTRICT77 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT78.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT78 OFF;
ALTER TABLE DISTRICT78 DROP CONSTRAINT
DISTRICT78CKC;
ALTER TABLE DISTRICT78 ADD CONSTRAINT
DISTRICT78CKC CHECK (D_W_ID BETWEEN 92401 AND
93600);
SET INTEGRITY FOR DISTRICT78 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT79.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT79 OFF;
ALTER TABLE DISTRICT79 DROP CONSTRAINT
DISTRICT79CKC;
ALTER TABLE DISTRICT79 ADD CONSTRAINT
DISTRICT79CKC CHECK (D_W_ID BETWEEN 93601 AND
94800);
SET INTEGRITY FOR DISTRICT79 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 7201 AND
8400);
SET INTEGRITY FOR DISTRICT7 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_DISTRICT80.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT80 OFF;
ALTER TABLE DISTRICT80 DROP CONSTRAINT
DISTRICT80CKC;
ALTER TABLE DISTRICT80 ADD CONSTRAINT
DISTRICT80CKC CHECK (D_W_ID >= 94801);
SET INTEGRITY FOR DISTRICT80 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 8401 AND
9600);
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 9601 AND
10800);
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 10801 AND
12000);
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 12001 AND
13200);
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 13201 AND
14400);
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 14401 AND
15600);
```

```
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 15601 AND
16800);
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 16801 AND
18000);
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 18001 AND
19200);
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 19201 AND
20400);
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 20401 AND
21600);
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 21601 AND
22800);
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 1200);
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 22801 AND
24000);
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 24001 AND
25200);
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 25201 AND
26400);
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 26401 AND
27600);
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 27601 AND 28800);
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 28801 AND 30000);
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 30001 AND 31200);
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 31201 AND 32400);

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 32401 AND 33600);
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 33601 AND 34800);
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 1201 AND 2400);
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 34801 AND 36000);
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 36001 AND 37200);
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 BETWEEN 37201 AND 38400);
SET INTEGRITY FOR HISTORY32 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY33.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY33 OFF;
ALTER TABLE HISTORY33 DROP CONSTRAINT HISTORY33CKC;
ALTER TABLE HISTORY33 ADD CONSTRAINT HISTORY33CKC CHECK (H_W_ID BETWEEN 38401 AND 39600);
SET INTEGRITY FOR HISTORY33 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY34.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY34 OFF;
ALTER TABLE HISTORY34 DROP CONSTRAINT
HISTORY34CKC;
ALTER TABLE HISTORY34 ADD CONSTRAINT
HISTORY34CKC CHECK (H_W_ID BETWEEN 39601 AND
40800);
SET INTEGRITY FOR HISTORY34 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY35.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY35 OFF;
ALTER TABLE HISTORY35 DROP CONSTRAINT
HISTORY35CKC;
ALTER TABLE HISTORY35 ADD CONSTRAINT
HISTORY35CKC CHECK (H_W_ID BETWEEN 40801 AND
42000);
SET INTEGRITY FOR HISTORY35 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY36.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY36 OFF;
ALTER TABLE HISTORY36 DROP CONSTRAINT
HISTORY36CKC;
ALTER TABLE HISTORY36 ADD CONSTRAINT
HISTORY36CKC CHECK (H_W_ID BETWEEN 42001 AND
43200);
SET INTEGRITY FOR HISTORY36 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY37.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY37 OFF;
ALTER TABLE HISTORY37 DROP CONSTRAINT
HISTORY37CKC;
ALTER TABLE HISTORY37 ADD CONSTRAINT
HISTORY37CKC CHECK (H_W_ID BETWEEN 43201 AND
44400);
SET INTEGRITY FOR HISTORY37 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY38.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY38 OFF;
ALTER TABLE HISTORY38 DROP CONSTRAINT
HISTORY38CKC;
ALTER TABLE HISTORY38 ADD CONSTRAINT
HISTORY38CKC CHECK (H_W_ID BETWEEN 44401 AND
45600);
SET INTEGRITY FOR HISTORY38 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY39.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY39 OFF;
ALTER TABLE HISTORY39 DROP CONSTRAINT
HISTORY39CKC;
ALTER TABLE HISTORY39 ADD CONSTRAINT
HISTORY39CKC CHECK (H_W_ID BETWEEN 45601 AND
46800);
SET INTEGRITY FOR HISTORY39 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 2401 AND
3600);
SET INTEGRITY FOR HISTORY3 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY40.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY40 OFF;
ALTER TABLE HISTORY40 DROP CONSTRAINT
HISTORY40CKC;
ALTER TABLE HISTORY40 ADD CONSTRAINT
HISTORY40CKC CHECK (H_W_ID BETWEEN 46801 AND
48000);

SET INTEGRITY FOR HISTORY40 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY41.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY41 OFF;
ALTER TABLE HISTORY41 DROP CONSTRAINT
HISTORY41CKC;
ALTER TABLE HISTORY41 ADD CONSTRAINT
HISTORY41CKC CHECK (H_W_ID BETWEEN 48001 AND
49200);
SET INTEGRITY FOR HISTORY41 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY42.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY42 OFF;
ALTER TABLE HISTORY42 DROP CONSTRAINT
HISTORY42CKC;
ALTER TABLE HISTORY42 ADD CONSTRAINT
HISTORY42CKC CHECK (H_W_ID BETWEEN 49201 AND
50400);
SET INTEGRITY FOR HISTORY42 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY43.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY43 OFF;
ALTER TABLE HISTORY43 DROP CONSTRAINT
HISTORY43CKC;
ALTER TABLE HISTORY43 ADD CONSTRAINT
HISTORY43CKC CHECK (H_W_ID BETWEEN 50401 AND
51600);
SET INTEGRITY FOR HISTORY43 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY44.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY44 OFF;

```
ALTER TABLE HISTORY44 DROP CONSTRAINT
HISTORY44CKC;
ALTER TABLE HISTORY44 ADD CONSTRAINT
HISTORY44CKC CHECK (H_W_ID BETWEEN 51601 AND
52800);
SET INTEGRITY FOR HISTORY44 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY45.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY45 OFF;
ALTER TABLE HISTORY45 DROP CONSTRAINT
HISTORY45CKC;
ALTER TABLE HISTORY45 ADD CONSTRAINT
HISTORY45CKC CHECK (H_W_ID BETWEEN 52801 AND
54000);
SET INTEGRITY FOR HISTORY45 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY46.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY46 OFF;
ALTER TABLE HISTORY46 DROP CONSTRAINT
HISTORY46CKC;
ALTER TABLE HISTORY46 ADD CONSTRAINT
HISTORY46CKC CHECK (H_W_ID BETWEEN 54001 AND
55200);
SET INTEGRITY FOR HISTORY46 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY47.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY47 OFF;
ALTER TABLE HISTORY47 DROP CONSTRAINT
HISTORY47CKC;
ALTER TABLE HISTORY47 ADD CONSTRAINT
HISTORY47CKC CHECK (H_W_ID BETWEEN 55201 AND
56400);
SET INTEGRITY FOR HISTORY47 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY48.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY48 OFF;
ALTER TABLE HISTORY48 DROP CONSTRAINT
HISTORY48CKC;
ALTER TABLE HISTORY48 ADD CONSTRAINT
HISTORY48CKC CHECK (H_W_ID BETWEEN 56401 AND
57600);
SET INTEGRITY FOR HISTORY48 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY49.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY49 OFF;
ALTER TABLE HISTORY49 DROP CONSTRAINT
HISTORY49CKC;
ALTER TABLE HISTORY49 ADD CONSTRAINT
HISTORY49CKC CHECK (H_W_ID BETWEEN 57601 AND
58800);
SET INTEGRITY FOR HISTORY49 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 3601 AND
4800);
SET INTEGRITY FOR HISTORY4 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY50.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY50 OFF;
ALTER TABLE HISTORY50 DROP CONSTRAINT
HISTORY50CKC;
ALTER TABLE HISTORY50 ADD CONSTRAINT
HISTORY50CKC CHECK (H_W_ID BETWEEN 58801 AND
60000);
SET INTEGRITY FOR HISTORY50 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY51.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY51 OFF;
ALTER TABLE HISTORY51 DROP CONSTRAINT
HISTORY51CKC;
ALTER TABLE HISTORY51 ADD CONSTRAINT
HISTORY51CKC CHECK (H_W_ID BETWEEN 60001 AND
61200);
SET INTEGRITY FOR HISTORY51 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY52.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY52 OFF;
ALTER TABLE HISTORY52 DROP CONSTRAINT
HISTORY52CKC;
ALTER TABLE HISTORY52 ADD CONSTRAINT
HISTORY52CKC CHECK (H_W_ID BETWEEN 61201 AND
62400);
SET INTEGRITY FOR HISTORY52 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY53.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY53 OFF;
ALTER TABLE HISTORY53 DROP CONSTRAINT
HISTORY53CKC;
ALTER TABLE HISTORY53 ADD CONSTRAINT
HISTORY53CKC CHECK (H_W_ID BETWEEN 62401 AND
63600);
SET INTEGRITY FOR HISTORY53 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST_HISTORY54.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY54 OFF;
ALTER TABLE HISTORY54 DROP CONSTRAINT
HISTORY54CKC;
ALTER TABLE HISTORY54 ADD CONSTRAINT
HISTORY54CKC CHECK (H_W_ID BETWEEN 63601 AND
64800);
```

```
SET INTEGRITY FOR HISTORY54 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY55.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY55 OFF;
ALTER TABLE HISTORY55 DROP CONSTRAINT
HISTORY55CKC;
ALTER TABLE HISTORY55 ADD CONSTRAINT
HISTORY55CKC CHECK (H_W_ID BETWEEN 64801 AND
66000);
SET INTEGRITY FOR HISTORY55 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY56.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY56 OFF;
ALTER TABLE HISTORY56 DROP CONSTRAINT
HISTORY56CKC;
ALTER TABLE HISTORY56 ADD CONSTRAINT
HISTORY56CKC CHECK (H_W_ID BETWEEN 66001 AND
67200);
SET INTEGRITY FOR HISTORY56 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY57.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY57 OFF;
ALTER TABLE HISTORY57 DROP CONSTRAINT
HISTORY57CKC;
ALTER TABLE HISTORY57 ADD CONSTRAINT
HISTORY57CKC CHECK (H_W_ID BETWEEN 67201 AND
68400);
SET INTEGRITY FOR HISTORY57 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY58.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY58 OFF;
```

```
ALTER TABLE HISTORY58 DROP CONSTRAINT
HISTORY58CKC;
ALTER TABLE HISTORY58 ADD CONSTRAINT
HISTORY58CKC CHECK (H_W_ID BETWEEN 68401 AND
69600);
SET INTEGRITY FOR HISTORY58 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY59.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY59 OFF;
ALTER TABLE HISTORY59 DROP CONSTRAINT
HISTORY59CKC;
ALTER TABLE HISTORY59 ADD CONSTRAINT
HISTORY59CKC CHECK (H_W_ID BETWEEN 69601 AND
70800);
SET INTEGRITY FOR HISTORY59 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 4801 AND
6000);
SET INTEGRITY FOR HISTORY5 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY60.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY60 OFF;
ALTER TABLE HISTORY60 DROP CONSTRAINT
HISTORY60CKC;
ALTER TABLE HISTORY60 ADD CONSTRAINT
HISTORY60CKC CHECK (H_W_ID BETWEEN 70801 AND
72000);
SET INTEGRITY FOR HISTORY60 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY61.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY61 OFF;
ALTER TABLE HISTORY61 DROP CONSTRAINT
HISTORY61CKC;
ALTER TABLE HISTORY61 ADD CONSTRAINT
HISTORY61CKC CHECK (H_W_ID BETWEEN 72001 AND
73200);
SET INTEGRITY FOR HISTORY61 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY62.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY62 OFF;
ALTER TABLE HISTORY62 DROP CONSTRAINT
HISTORY62CKC;
ALTER TABLE HISTORY62 ADD CONSTRAINT
HISTORY62CKC CHECK (H_W_ID BETWEEN 73201 AND
74400);
SET INTEGRITY FOR HISTORY62 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY63.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY63 OFF;
ALTER TABLE HISTORY63 DROP CONSTRAINT
HISTORY63CKC;
ALTER TABLE HISTORY63 ADD CONSTRAINT
HISTORY63CKC CHECK (H_W_ID BETWEEN 74401 AND
75600);
SET INTEGRITY FOR HISTORY63 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY64.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY64 OFF;
ALTER TABLE HISTORY64 DROP CONSTRAINT
HISTORY64CKC;
ALTER TABLE HISTORY64 ADD CONSTRAINT
HISTORY64CKC CHECK (H_W_ID BETWEEN 75601 AND
76800);
SET INTEGRITY FOR HISTORY64 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST HISTORY65.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY65 OFF;
ALTER TABLE HISTORY65 DROP CONSTRAINT HISTORY65CKC;
ALTER TABLE HISTORY65 ADD CONSTRAINT HISTORY65CKC CHECK (H_W_ID BETWEEN 76801 AND 78000);
SET INTEGRITY FOR HISTORY65 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY66.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY66 OFF;
ALTER TABLE HISTORY66 DROP CONSTRAINT HISTORY66CKC;
ALTER TABLE HISTORY66 ADD CONSTRAINT HISTORY66CKC CHECK (H_W_ID BETWEEN 78001 AND 79200);
SET INTEGRITY FOR HISTORY66 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY67.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY67 OFF;
ALTER TABLE HISTORY67 DROP CONSTRAINT HISTORY67CKC;
ALTER TABLE HISTORY67 ADD CONSTRAINT HISTORY67CKC CHECK (H_W_ID BETWEEN 79201 AND 80400);
SET INTEGRITY FOR HISTORY67 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY68.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY68 OFF;
ALTER TABLE HISTORY68 DROP CONSTRAINT HISTORY68CKC;
ALTER TABLE HISTORY68 ADD CONSTRAINT HISTORY68CKC CHECK (H_W_ID BETWEEN 80401 AND 81600);

SET INTEGRITY FOR HISTORY68 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY69.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY69 OFF;
ALTER TABLE HISTORY69 DROP CONSTRAINT HISTORY69CKC;
ALTER TABLE HISTORY69 ADD CONSTRAINT HISTORY69CKC CHECK (H_W_ID BETWEEN 81601 AND 82800);
SET INTEGRITY FOR HISTORY69 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 6001 AND 7200);
SET INTEGRITY FOR HISTORY6 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY70.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY70 OFF;
ALTER TABLE HISTORY70 DROP CONSTRAINT HISTORY70CKC;
ALTER TABLE HISTORY70 ADD CONSTRAINT HISTORY70CKC CHECK (H_W_ID BETWEEN 82801 AND 84000);
SET INTEGRITY FOR HISTORY70 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY71.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY71 OFF;

ALTER TABLE HISTORY71 DROP CONSTRAINT HISTORY71CKC;
ALTER TABLE HISTORY71 ADD CONSTRAINT HISTORY71CKC CHECK (H_W_ID BETWEEN 84001 AND 85200);
SET INTEGRITY FOR HISTORY71 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY72.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY72 OFF;
ALTER TABLE HISTORY72 DROP CONSTRAINT HISTORY72CKC;
ALTER TABLE HISTORY72 ADD CONSTRAINT HISTORY72CKC CHECK (H_W_ID BETWEEN 85201 AND 86400);
SET INTEGRITY FOR HISTORY72 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY73.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY73 OFF;
ALTER TABLE HISTORY73 DROP CONSTRAINT HISTORY73CKC;
ALTER TABLE HISTORY73 ADD CONSTRAINT HISTORY73CKC CHECK (H_W_ID BETWEEN 86401 AND 87600);
SET INTEGRITY FOR HISTORY73 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY74.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY74 OFF;
ALTER TABLE HISTORY74 DROP CONSTRAINT HISTORY74CKC;
ALTER TABLE HISTORY74 ADD CONSTRAINT HISTORY74CKC CHECK (H_W_ID BETWEEN 87601 AND 88800);
SET INTEGRITY FOR HISTORY74 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST HISTORY75.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY75 OFF;
ALTER TABLE HISTORY75 DROP CONSTRAINT
HISTORY75CKC;
ALTER TABLE HISTORY75 ADD CONSTRAINT
HISTORY75CKC CHECK (H_W_ID BETWEEN 88801 AND
90000);
SET INTEGRITY FOR HISTORY75 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY76.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY76 OFF;
ALTER TABLE HISTORY76 DROP CONSTRAINT
HISTORY76CKC;
ALTER TABLE HISTORY76 ADD CONSTRAINT
HISTORY76CKC CHECK (H_W_ID BETWEEN 90001 AND
91200);
SET INTEGRITY FOR HISTORY76 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY77.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY77 OFF;
ALTER TABLE HISTORY77 DROP CONSTRAINT
HISTORY77CKC;
ALTER TABLE HISTORY77 ADD CONSTRAINT
HISTORY77CKC CHECK (H_W_ID BETWEEN 91201 AND
92400);
SET INTEGRITY FOR HISTORY77 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY78.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY78 OFF;
ALTER TABLE HISTORY78 DROP CONSTRAINT
HISTORY78CKC;
ALTER TABLE HISTORY78 ADD CONSTRAINT
HISTORY78CKC CHECK (H_W_ID BETWEEN 92401 AND
93600);
SET INTEGRITY FOR HISTORY78 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY79.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY79 OFF;
ALTER TABLE HISTORY79 DROP CONSTRAINT
HISTORY79CKC;
ALTER TABLE HISTORY79 ADD CONSTRAINT
HISTORY79CKC CHECK (H_W_ID BETWEEN 93601 AND
94800);
SET INTEGRITY FOR HISTORY79 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 7201 AND
8400);
SET INTEGRITY FOR HISTORY7 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST_HISTORY80.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY80 OFF;
ALTER TABLE HISTORY80 DROP CONSTRAINT
HISTORY80CKC;
ALTER TABLE HISTORY80 ADD CONSTRAINT
HISTORY80CKC CHECK (H_W_ID >= 94801);
SET INTEGRITY FOR HISTORY80 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 8401 AND
9600);
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 9601 AND
10800);
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
10801 AND 12000) 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
12001 AND 13200) 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
13201 AND 14400) 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
14401 AND 15600) 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
15601 AND 16800) 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
16801 AND 18000) 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
18001 AND 19200) 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
19201 AND 20400) 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
20401 AND 21600) 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
21601 AND 22800) 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 1200) 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
22801 AND 24000) 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
24001 AND 25200) 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
25201 AND 26400) 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
26401 AND 27600) 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
27601 AND 28800) 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
28801 AND 30000) 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
30001 AND 31200) 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
31201 AND 32400) 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
32401 AND 33600) 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
33601 AND 34800) 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 1201
AND 2400) 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
34801 AND 36000) 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
36001 AND 37200) 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 BETWEEN
37201 AND 38400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA32 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA33.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA33 OFF;
ALTER TABLE NEW_ORDERA33 DROP CONSTRAINT
NEW_ORDERA33CKC;
ALTER TABLE NEW_ORDERA33 ADD CONSTRAINT
NEW_ORDERA33CKC CHECK ((NO_W_ID BETWEEN
38401 AND 39600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA33 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA34.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA34 OFF;
ALTER TABLE NEW_ORDERA34 DROP CONSTRAINT
NEW_ORDERA34CKC;
ALTER TABLE NEW_ORDERA34 ADD CONSTRAINT
NEW_ORDERA34CKC CHECK ((NO_W_ID BETWEEN
39601 AND 40800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA34 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA35.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA35 OFF;
ALTER TABLE NEW_ORDERA35 DROP CONSTRAINT
NEW_ORDERA35CKC;
ALTER TABLE NEW_ORDERA35 ADD CONSTRAINT
NEW_ORDERA35CKC CHECK ((NO_W_ID BETWEEN
40801 AND 42000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA35 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA36.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA36 OFF;
ALTER TABLE NEW_ORDERA36 DROP CONSTRAINT
NEW_ORDERA36CKC;
ALTER TABLE NEW_ORDERA36 ADD CONSTRAINT
NEW_ORDERA36CKC CHECK ((NO_W_ID BETWEEN
42001 AND 43200) AND (NO_O_ID <= 3675));

SET INTEGRITY FOR NEW_ORDERA36 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA37.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA37 OFF;
ALTER TABLE NEW_ORDERA37 DROP CONSTRAINT
NEW_ORDERA37CKC;
ALTER TABLE NEW_ORDERA37 ADD CONSTRAINT
NEW_ORDERA37CKC CHECK ((NO_W_ID BETWEEN
43201 AND 44400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA37 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA38.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA38 OFF;
ALTER TABLE NEW_ORDERA38 DROP CONSTRAINT
NEW_ORDERA38CKC;
ALTER TABLE NEW_ORDERA38 ADD CONSTRAINT
NEW_ORDERA38CKC CHECK ((NO_W_ID BETWEEN
44401 AND 45600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA38 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA39.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA39 OFF;
ALTER TABLE NEW_ORDERA39 DROP CONSTRAINT
NEW_ORDERA39CKC;
ALTER TABLE NEW_ORDERA39 ADD CONSTRAINT
NEW_ORDERA39CKC CHECK ((NO_W_ID BETWEEN
45601 AND 46800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA39 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 2401
AND 3600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA3 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA40.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA40 OFF;
ALTER TABLE NEW_ORDERA40 DROP CONSTRAINT
NEW_ORDERA40CKC;
ALTER TABLE NEW_ORDERA40 ADD CONSTRAINT
NEW_ORDERA40CKC CHECK ((NO_W_ID BETWEEN
46801 AND 48000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA40 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA41.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA41 OFF;
ALTER TABLE NEW_ORDERA41 DROP CONSTRAINT
NEW_ORDERA41CKC;
ALTER TABLE NEW_ORDERA41 ADD CONSTRAINT
NEW_ORDERA41CKC CHECK ((NO_W_ID BETWEEN
48001 AND 49200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA41 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA42.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA42 OFF;
ALTER TABLE NEW_ORDERA42 DROP CONSTRAINT
NEW_ORDERA42CKC;
ALTER TABLE NEW_ORDERA42 ADD CONSTRAINT
NEW_ORDERA42CKC CHECK ((NO_W_ID BETWEEN
49201 AND 50400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA42 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA43.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA43 OFF;
ALTER TABLE NEW_ORDERA43 DROP CONSTRAINT
NEW_ORDERA43CKC;
ALTER TABLE NEW_ORDERA43 ADD CONSTRAINT
NEW_ORDERA43CKC CHECK ((NO_W_ID BETWEEN
50401 AND 51600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA43 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA44.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA44 OFF;
ALTER TABLE NEW_ORDERA44 DROP CONSTRAINT
NEW_ORDERA44CKC;
ALTER TABLE NEW_ORDERA44 ADD CONSTRAINT
NEW_ORDERA44CKC CHECK ((NO_W_ID BETWEEN
51601 AND 52800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA44 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA45.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA45 OFF;
ALTER TABLE NEW_ORDERA45 DROP CONSTRAINT
NEW_ORDERA45CKC;
ALTER TABLE NEW_ORDERA45 ADD CONSTRAINT
NEW_ORDERA45CKC CHECK ((NO_W_ID BETWEEN
52801 AND 54000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA45 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA46.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA46 OFF;
ALTER TABLE NEW_ORDERA46 DROP CONSTRAINT
NEW_ORDERA46CKC;
ALTER TABLE NEW_ORDERA46 ADD CONSTRAINT
NEW_ORDERA46CKC CHECK ((NO_W_ID BETWEEN
54001 AND 55200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA46 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA47.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA47 OFF;
ALTER TABLE NEW_ORDERA47 DROP CONSTRAINT
NEW_ORDERA47CKC;
ALTER TABLE NEW_ORDERA47 ADD CONSTRAINT
NEW_ORDERA47CKC CHECK ((NO_W_ID BETWEEN
55201 AND 56400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA47 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA48.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA48 OFF;
ALTER TABLE NEW_ORDERA48 DROP CONSTRAINT
NEW_ORDERA48CKC;
ALTER TABLE NEW_ORDERA48 ADD CONSTRAINT
NEW_ORDERA48CKC CHECK ((NO_W_ID BETWEEN
56401 AND 57600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA48 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA49.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA49 OFF;
ALTER TABLE NEW_ORDERA49 DROP CONSTRAINT
NEW_ORDERA49CKC;
ALTER TABLE NEW_ORDERA49 ADD CONSTRAINT
NEW_ORDERA49CKC CHECK ((NO_W_ID BETWEEN
57601 AND 58800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA49 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 3601
AND 4800) AND (NO_O_ID <= 3675));

SET INTEGRITY FOR NEW_ORDERA4 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA50.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA50 OFF;
ALTER TABLE NEW_ORDERA50 DROP CONSTRAINT
NEW_ORDERA50CKC;
ALTER TABLE NEW_ORDERA50 ADD CONSTRAINT
NEW_ORDERA50CKC CHECK ((NO_W_ID BETWEEN
58801 AND 60000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA50 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA51.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA51 OFF;
ALTER TABLE NEW_ORDERA51 DROP CONSTRAINT
NEW_ORDERA51CKC;
ALTER TABLE NEW_ORDERA51 ADD CONSTRAINT
NEW_ORDERA51CKC CHECK ((NO_W_ID BETWEEN
60001 AND 61200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA51 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA52.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA52 OFF;
ALTER TABLE NEW_ORDERA52 DROP CONSTRAINT
NEW_ORDERA52CKC;
ALTER TABLE NEW_ORDERA52 ADD CONSTRAINT
NEW_ORDERA52CKC CHECK ((NO_W_ID BETWEEN
61201 AND 62400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA52 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERA53.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA53 OFF;

```
ALTER TABLE NEW_ORDERA53 DROP CONSTRAINT
NEW_ORDERA53CKC;
ALTER TABLE NEW_ORDERA53 ADD CONSTRAINT
NEW_ORDERA53CKC CHECK ((NO_W_ID BETWEEN
62401 AND 63600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA53 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA54.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA54 OFF;
ALTER TABLE NEW_ORDERA54 DROP CONSTRAINT
NEW_ORDERA54CKC;
ALTER TABLE NEW_ORDERA54 ADD CONSTRAINT
NEW_ORDERA54CKC CHECK ((NO_W_ID BETWEEN
63601 AND 64800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA54 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA55.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA55 OFF;
ALTER TABLE NEW_ORDERA55 DROP CONSTRAINT
NEW_ORDERA55CKC;
ALTER TABLE NEW_ORDERA55 ADD CONSTRAINT
NEW_ORDERA55CKC CHECK ((NO_W_ID BETWEEN
64801 AND 66000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA55 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA56.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA56 OFF;
ALTER TABLE NEW_ORDERA56 DROP CONSTRAINT
NEW_ORDERA56CKC;
ALTER TABLE NEW_ORDERA56 ADD CONSTRAINT
NEW_ORDERA56CKC CHECK ((NO_W_ID BETWEEN
66001 AND 67200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA56 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA57.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA57 OFF;
ALTER TABLE NEW_ORDERA57 DROP CONSTRAINT
NEW_ORDERA57CKC;
ALTER TABLE NEW_ORDERA57 ADD CONSTRAINT
NEW_ORDERA57CKC CHECK ((NO_W_ID BETWEEN
67201 AND 68400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA57 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA58.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA58 OFF;
ALTER TABLE NEW_ORDERA58 DROP CONSTRAINT
NEW_ORDERA58CKC;
ALTER TABLE NEW_ORDERA58 ADD CONSTRAINT
NEW_ORDERA58CKC CHECK ((NO_W_ID BETWEEN
68401 AND 69600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA58 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA59.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA59 OFF;
ALTER TABLE NEW_ORDERA59 DROP CONSTRAINT
NEW_ORDERA59CKC;
ALTER TABLE NEW_ORDERA59 ADD CONSTRAINT
NEW_ORDERA59CKC CHECK ((NO_W_ID BETWEEN
69601 AND 70800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA59 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 4801
AND 6000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA5 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA60.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA60 OFF;
ALTER TABLE NEW_ORDERA60 DROP CONSTRAINT
NEW_ORDERA60CKC;
ALTER TABLE NEW_ORDERA60 ADD CONSTRAINT
NEW_ORDERA60CKC CHECK ((NO_W_ID BETWEEN
70801 AND 72000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA60 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA61.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA61 OFF;
ALTER TABLE NEW_ORDERA61 DROP CONSTRAINT
NEW_ORDERA61CKC;
ALTER TABLE NEW_ORDERA61 ADD CONSTRAINT
NEW_ORDERA61CKC CHECK ((NO_W_ID BETWEEN
72001 AND 73200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA61 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA62.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA62 OFF;
ALTER TABLE NEW_ORDERA62 DROP CONSTRAINT
NEW_ORDERA62CKC;
ALTER TABLE NEW_ORDERA62 ADD CONSTRAINT
NEW_ORDERA62CKC CHECK ((NO_W_ID BETWEEN
73201 AND 74400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA62 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW_ORDERA63.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA63 OFF;
ALTER TABLE NEW_ORDERA63 DROP CONSTRAINT
NEW_ORDERA63CKC;
ALTER TABLE NEW_ORDERA63 ADD CONSTRAINT
NEW_ORDERA63CKC CHECK ((NO_W_ID BETWEEN
74401 AND 75600) AND (NO_O_ID <= 3675));
```

```
SET INTEGRITY FOR NEW_ORDERA63 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA64.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA64 OFF;
ALTER TABLE NEW_ORDERA64 DROP CONSTRAINT
NEW_ORDERA64CKC;
ALTER TABLE NEW_ORDERA64 ADD CONSTRAINT
NEW_ORDERA64CKC CHECK ((NO_W_ID BETWEEN
75601 AND 76800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA64 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA65.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA65 OFF;
ALTER TABLE NEW_ORDERA65 DROP CONSTRAINT
NEW_ORDERA65CKC;
ALTER TABLE NEW_ORDERA65 ADD CONSTRAINT
NEW_ORDERA65CKC CHECK ((NO_W_ID BETWEEN
76801 AND 78000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA65 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA66.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA66 OFF;
ALTER TABLE NEW_ORDERA66 DROP CONSTRAINT
NEW_ORDERA66CKC;
ALTER TABLE NEW_ORDERA66 ADD CONSTRAINT
NEW_ORDERA66CKC CHECK ((NO_W_ID BETWEEN
78001 AND 79200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA66 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA67.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA67 OFF;
```

```
ALTER TABLE NEW_ORDERA67 DROP CONSTRAINT
NEW_ORDERA67CKC;
ALTER TABLE NEW_ORDERA67 ADD CONSTRAINT
NEW_ORDERA67CKC CHECK ((NO_W_ID BETWEEN
79201 AND 80400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA67 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA68.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA68 OFF;
ALTER TABLE NEW_ORDERA68 DROP CONSTRAINT
NEW_ORDERA68CKC;
ALTER TABLE NEW_ORDERA68 ADD CONSTRAINT
NEW_ORDERA68CKC CHECK ((NO_W_ID BETWEEN
80401 AND 81600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA68 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA69.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA69 OFF;
ALTER TABLE NEW_ORDERA69 DROP CONSTRAINT
NEW_ORDERA69CKC;
ALTER TABLE NEW_ORDERA69 ADD CONSTRAINT
NEW_ORDERA69CKC CHECK ((NO_W_ID BETWEEN
81601 AND 82800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA69 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 6001
AND 7200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA6 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA70.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA70 OFF;
ALTER TABLE NEW_ORDERA70 DROP CONSTRAINT
NEW_ORDERA70CKC;
ALTER TABLE NEW_ORDERA70 ADD CONSTRAINT
NEW_ORDERA70CKC CHECK ((NO_W_ID BETWEEN
82801 AND 84000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA70 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA71.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA71 OFF;
ALTER TABLE NEW_ORDERA71 DROP CONSTRAINT
NEW_ORDERA71CKC;
ALTER TABLE NEW_ORDERA71 ADD CONSTRAINT
NEW_ORDERA71CKC CHECK ((NO_W_ID BETWEEN
84001 AND 85200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA71 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA72.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA72 OFF;
ALTER TABLE NEW_ORDERA72 DROP CONSTRAINT
NEW_ORDERA72CKC;
ALTER TABLE NEW_ORDERA72 ADD CONSTRAINT
NEW_ORDERA72CKC CHECK ((NO_W_ID BETWEEN
85201 AND 86400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA72 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA73.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA73 OFF;
ALTER TABLE NEW_ORDERA73 DROP CONSTRAINT
NEW_ORDERA73CKC;
ALTER TABLE NEW_ORDERA73 ADD CONSTRAINT
NEW_ORDERA73CKC CHECK ((NO_W_ID BETWEEN
86401 AND 87600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA73 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA74.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA74 OFF;
ALTER TABLE NEW_ORDERA74 DROP CONSTRAINT
NEW_ORDERA74CKC;
ALTER TABLE NEW_ORDERA74 ADD CONSTRAINT
NEW_ORDERA74CKC CHECK ((NO_W_ID BETWEEN
87601 AND 88800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA74 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA75.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA75 OFF;
ALTER TABLE NEW_ORDERA75 DROP CONSTRAINT
NEW_ORDERA75CKC;
ALTER TABLE NEW_ORDERA75 ADD CONSTRAINT
NEW_ORDERA75CKC CHECK ((NO_W_ID BETWEEN
88801 AND 90000) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA75 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA76.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA76 OFF;
ALTER TABLE NEW_ORDERA76 DROP CONSTRAINT
NEW_ORDERA76CKC;
ALTER TABLE NEW_ORDERA76 ADD CONSTRAINT
NEW_ORDERA76CKC CHECK ((NO_W_ID BETWEEN
90001 AND 91200) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA76 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA77.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA77 OFF;
ALTER TABLE NEW_ORDERA77 DROP CONSTRAINT
NEW_ORDERA77CKC;
ALTER TABLE NEW_ORDERA77 ADD CONSTRAINT
NEW_ORDERA77CKC CHECK ((NO_W_ID BETWEEN
91201 AND 92400) AND (NO_O_ID <= 3675));
```

```
SET INTEGRITY FOR NEW_ORDERA77 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA78.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA78 OFF;
ALTER TABLE NEW_ORDERA78 DROP CONSTRAINT
NEW_ORDERA78CKC;
ALTER TABLE NEW_ORDERA78 ADD CONSTRAINT
NEW_ORDERA78CKC CHECK ((NO_W_ID BETWEEN
92401 AND 93600) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA78 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA79.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA79 OFF;
ALTER TABLE NEW_ORDERA79 DROP CONSTRAINT
NEW_ORDERA79CKC;
ALTER TABLE NEW_ORDERA79 ADD CONSTRAINT
NEW_ORDERA79CKC CHECK ((NO_W_ID BETWEEN
93601 AND 94800) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA79 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 7201
AND 8400) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERA80.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA80 OFF;
```

```
ALTER TABLE NEW_ORDERA80 DROP CONSTRAINT
NEW_ORDERA80CKC;
ALTER TABLE NEW_ORDERA80 ADD CONSTRAINT
NEW_ORDERA80CKC CHECK ((NO_W_ID >= 94801) AND
(NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDERA80 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 8401
AND 9600) 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 9601
AND 10800) 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
10801 AND 12000) 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
12001 AND 13200) 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
13201 AND 14400) 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
14401 AND 15600) 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
15601 AND 16800) 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
16801 AND 18000) 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
18001 AND 19200) 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
19201 AND 20400) 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
20401 AND 21600) 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
21601 AND 22800) 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 1200) 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
22801 AND 24000) 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
24001 AND 25200) 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
25201 AND 26400) 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
26401 AND 27600) 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
27601 AND 28800) 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
28801 AND 30000) 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
30001 AND 31200) 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
31201 AND 32400) 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
32401 AND 33600) 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
33601 AND 34800) 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 1201
AND 2400) 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
34801 AND 36000) 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
36001 AND 37200) 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 BETWEEN
37201 AND 38400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB32 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB33.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB33 OFF;
ALTER TABLE NEW_ORDERB33 DROP CONSTRAINT
NEW_ORDERB33CKC;
ALTER TABLE NEW_ORDERB33 ADD CONSTRAINT
NEW_ORDERB33CKC CHECK ((NO_W_ID BETWEEN
38401 AND 39600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB33 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB34.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB34 OFF;
ALTER TABLE NEW_ORDERB34 DROP CONSTRAINT
NEW_ORDERB34CKC;
ALTER TABLE NEW_ORDERB34 ADD CONSTRAINT
NEW_ORDERB34CKC CHECK ((NO_W_ID BETWEEN
39601 AND 40800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB34 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB35.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB35 OFF;
```

```
ALTER TABLE NEW_ORDERB35 DROP CONSTRAINT
NEW_ORDERB35CKC;
ALTER TABLE NEW_ORDERB35 ADD CONSTRAINT
NEW_ORDERB35CKC CHECK ((NO_W_ID BETWEEN
40801 AND 42000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB35 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB36.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB36 OFF;
ALTER TABLE NEW_ORDERB36 DROP CONSTRAINT
NEW_ORDERB36CKC;
ALTER TABLE NEW_ORDERB36 ADD CONSTRAINT
NEW_ORDERB36CKC CHECK ((NO_W_ID BETWEEN
42001 AND 43200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB36 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB37.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB37 OFF;
ALTER TABLE NEW_ORDERB37 DROP CONSTRAINT
NEW_ORDERB37CKC;
ALTER TABLE NEW_ORDERB37 ADD CONSTRAINT
NEW_ORDERB37CKC CHECK ((NO_W_ID BETWEEN
43201 AND 44400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB37 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB38.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB38 OFF;
ALTER TABLE NEW_ORDERB38 DROP CONSTRAINT
NEW_ORDERB38CKC;
ALTER TABLE NEW_ORDERB38 ADD CONSTRAINT
NEW_ORDERB38CKC CHECK ((NO_W_ID BETWEEN
44401 AND 45600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB38 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB39.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB39 OFF;
ALTER TABLE NEW_ORDERB39 DROP CONSTRAINT
NEW_ORDERB39CKC;
ALTER TABLE NEW_ORDERB39 ADD CONSTRAINT
NEW_ORDERB39CKC CHECK ((NO_W_ID BETWEEN
45601 AND 46800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB39 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 2401
AND 3600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB3 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB40.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB40 OFF;
ALTER TABLE NEW_ORDERB40 DROP CONSTRAINT
NEW_ORDERB40CKC;
ALTER TABLE NEW_ORDERB40 ADD CONSTRAINT
NEW_ORDERB40CKC CHECK ((NO_W_ID BETWEEN
46801 AND 48000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB40 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB41.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB41 OFF;
ALTER TABLE NEW_ORDERB41 DROP CONSTRAINT
NEW_ORDERB41CKC;
ALTER TABLE NEW_ORDERB41 ADD CONSTRAINT
NEW_ORDERB41CKC CHECK ((NO_W_ID BETWEEN
48001 AND 49200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB41 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB42.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB42 OFF;
ALTER TABLE NEW_ORDERB42 DROP CONSTRAINT
NEW_ORDERB42CKC;
ALTER TABLE NEW_ORDERB42 ADD CONSTRAINT
NEW_ORDERB42CKC CHECK ((NO_W_ID BETWEEN
49201 AND 50400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB42 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB43.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB43 OFF;
ALTER TABLE NEW_ORDERB43 DROP CONSTRAINT
NEW_ORDERB43CKC;
ALTER TABLE NEW_ORDERB43 ADD CONSTRAINT
NEW_ORDERB43CKC CHECK ((NO_W_ID BETWEEN
50401 AND 51600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB43 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB44.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB44 OFF;
ALTER TABLE NEW_ORDERB44 DROP CONSTRAINT
NEW_ORDERB44CKC;
ALTER TABLE NEW_ORDERB44 ADD CONSTRAINT
NEW_ORDERB44CKC CHECK ((NO_W_ID BETWEEN
51601 AND 52800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB44 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB45.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB45 OFF;
ALTER TABLE NEW_ORDERB45 DROP CONSTRAINT
NEW_ORDERB45CKC;
ALTER TABLE NEW_ORDERB45 ADD CONSTRAINT
NEW_ORDERB45CKC CHECK ((NO_W_ID BETWEEN
52801 AND 54000) AND (NO_O_ID >= 3676));

SET INTEGRITY FOR NEW_ORDERB45 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB46.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB46 OFF;
ALTER TABLE NEW_ORDERB46 DROP CONSTRAINT
NEW_ORDERB46CKC;
ALTER TABLE NEW_ORDERB46 ADD CONSTRAINT
NEW_ORDERB46CKC CHECK ((NO_W_ID BETWEEN
54001 AND 55200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB46 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB47.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB47 OFF;
ALTER TABLE NEW_ORDERB47 DROP CONSTRAINT
NEW_ORDERB47CKC;
ALTER TABLE NEW_ORDERB47 ADD CONSTRAINT
NEW_ORDERB47CKC CHECK ((NO_W_ID BETWEEN
55201 AND 56400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB47 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB48.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB48 OFF;
ALTER TABLE NEW_ORDERB48 DROP CONSTRAINT
NEW_ORDERB48CKC;
ALTER TABLE NEW_ORDERB48 ADD CONSTRAINT
NEW_ORDERB48CKC CHECK ((NO_W_ID BETWEEN
56401 AND 57600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB48 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB49.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB49 OFF;

ALTER TABLE NEW_ORDERB49 DROP CONSTRAINT
NEW_ORDERB49CKC;
ALTER TABLE NEW_ORDERB49 ADD CONSTRAINT
NEW_ORDERB49CKC CHECK ((NO_W_ID BETWEEN
57601 AND 58800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB49 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 3601
AND 4800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB4 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB50.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB50 OFF;
ALTER TABLE NEW_ORDERB50 DROP CONSTRAINT
NEW_ORDERB50CKC;
ALTER TABLE NEW_ORDERB50 ADD CONSTRAINT
NEW_ORDERB50CKC CHECK ((NO_W_ID BETWEEN
58801 AND 60000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB50 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB51.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB51 OFF;
ALTER TABLE NEW_ORDERB51 DROP CONSTRAINT
NEW_ORDERB51CKC;
ALTER TABLE NEW_ORDERB51 ADD CONSTRAINT
NEW_ORDERB51CKC CHECK ((NO_W_ID BETWEEN
60001 AND 61200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB51 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST NEW ORDERB52.ddl


```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB52 OFF;
ALTER TABLE NEW_ORDERB52 DROP CONSTRAINT
NEW_ORDERB52CKC;
ALTER TABLE NEW_ORDERB52 ADD CONSTRAINT
NEW_ORDERB52CKC CHECK ((NO_W_ID BETWEEN
61201 AND 62400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB52 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB53.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB53 OFF;
ALTER TABLE NEW_ORDERB53 DROP CONSTRAINT
NEW_ORDERB53CKC;
ALTER TABLE NEW_ORDERB53 ADD CONSTRAINT
NEW_ORDERB53CKC CHECK ((NO_W_ID BETWEEN
62401 AND 63600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB53 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB54.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB54 OFF;
ALTER TABLE NEW_ORDERB54 DROP CONSTRAINT
NEW_ORDERB54CKC;
ALTER TABLE NEW_ORDERB54 ADD CONSTRAINT
NEW_ORDERB54CKC CHECK ((NO_W_ID BETWEEN
63601 AND 64800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB54 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB55.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB55 OFF;
ALTER TABLE NEW_ORDERB55 DROP CONSTRAINT
NEW_ORDERB55CKC;
ALTER TABLE NEW_ORDERB55 ADD CONSTRAINT
NEW_ORDERB55CKC CHECK ((NO_W_ID BETWEEN
64801 AND 66000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB55 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB56.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB56 OFF;
ALTER TABLE NEW_ORDERB56 DROP CONSTRAINT
NEW_ORDERB56CKC;
ALTER TABLE NEW_ORDERB56 ADD CONSTRAINT
NEW_ORDERB56CKC CHECK ((NO_W_ID BETWEEN
66001 AND 67200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB56 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB57.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB57 OFF;
ALTER TABLE NEW_ORDERB57 DROP CONSTRAINT
NEW_ORDERB57CKC;
ALTER TABLE NEW_ORDERB57 ADD CONSTRAINT
NEW_ORDERB57CKC CHECK ((NO_W_ID BETWEEN
67201 AND 68400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB57 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB58.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB58 OFF;
ALTER TABLE NEW_ORDERB58 DROP CONSTRAINT
NEW_ORDERB58CKC;
ALTER TABLE NEW_ORDERB58 ADD CONSTRAINT
NEW_ORDERB58CKC CHECK ((NO_W_ID BETWEEN
68401 AND 69600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB58 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB59.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB59 OFF;
ALTER TABLE NEW_ORDERB59 DROP CONSTRAINT
NEW_ORDERB59CKC;
ALTER TABLE NEW_ORDERB59 ADD CONSTRAINT
NEW_ORDERB59CKC CHECK ((NO_W_ID BETWEEN
69601 AND 70800) AND (NO_O_ID >= 3676));
```

```
SET INTEGRITY FOR NEW_ORDERB59 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 4801
AND 6000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB5 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB60.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB60 OFF;
ALTER TABLE NEW_ORDERB60 DROP CONSTRAINT
NEW_ORDERB60CKC;
ALTER TABLE NEW_ORDERB60 ADD CONSTRAINT
NEW_ORDERB60CKC CHECK ((NO_W_ID BETWEEN
70801 AND 72000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB60 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB61.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB61 OFF;
ALTER TABLE NEW_ORDERB61 DROP CONSTRAINT
NEW_ORDERB61CKC;
ALTER TABLE NEW_ORDERB61 ADD CONSTRAINT
NEW_ORDERB61CKC CHECK ((NO_W_ID BETWEEN
72001 AND 73200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB61 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB62.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB62 OFF;
```

```
ALTER TABLE NEW_ORDERB62 DROP CONSTRAINT
NEW_ORDERB62CKC;
ALTER TABLE NEW_ORDERB62 ADD CONSTRAINT
NEW_ORDERB62CKC CHECK ((NO_W_ID BETWEEN
73201 AND 74400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB62 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB63.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB63 OFF;
ALTER TABLE NEW_ORDERB63 DROP CONSTRAINT
NEW_ORDERB63CKC;
ALTER TABLE NEW_ORDERB63 ADD CONSTRAINT
NEW_ORDERB63CKC CHECK ((NO_W_ID BETWEEN
74401 AND 75600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB63 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB64.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB64 OFF;
ALTER TABLE NEW_ORDERB64 DROP CONSTRAINT
NEW_ORDERB64CKC;
ALTER TABLE NEW_ORDERB64 ADD CONSTRAINT
NEW_ORDERB64CKC CHECK ((NO_W_ID BETWEEN
75601 AND 76800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB64 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB65.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB65 OFF;
ALTER TABLE NEW_ORDERB65 DROP CONSTRAINT
NEW_ORDERB65CKC;
ALTER TABLE NEW_ORDERB65 ADD CONSTRAINT
NEW_ORDERB65CKC CHECK ((NO_W_ID BETWEEN
76801 AND 78000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB65 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB66.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB66 OFF;
ALTER TABLE NEW_ORDERB66 DROP CONSTRAINT
NEW_ORDERB66CKC;
ALTER TABLE NEW_ORDERB66 ADD CONSTRAINT
NEW_ORDERB66CKC CHECK ((NO_W_ID BETWEEN
78001 AND 79200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB66 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB67.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB67 OFF;
ALTER TABLE NEW_ORDERB67 DROP CONSTRAINT
NEW_ORDERB67CKC;
ALTER TABLE NEW_ORDERB67 ADD CONSTRAINT
NEW_ORDERB67CKC CHECK ((NO_W_ID BETWEEN
79201 AND 80400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB67 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB68.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB68 OFF;
ALTER TABLE NEW_ORDERB68 DROP CONSTRAINT
NEW_ORDERB68CKC;
ALTER TABLE NEW_ORDERB68 ADD CONSTRAINT
NEW_ORDERB68CKC CHECK ((NO_W_ID BETWEEN
80401 AND 81600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB68 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB69.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB69 OFF;
ALTER TABLE NEW_ORDERB69 DROP CONSTRAINT
NEW_ORDERB69CKC;
ALTER TABLE NEW_ORDERB69 ADD CONSTRAINT
NEW_ORDERB69CKC CHECK ((NO_W_ID BETWEEN
81601 AND 82800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB69 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 6001
AND 7200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB6 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB70.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB70 OFF;
ALTER TABLE NEW_ORDERB70 DROP CONSTRAINT
NEW_ORDERB70CKC;
ALTER TABLE NEW_ORDERB70 ADD CONSTRAINT
NEW_ORDERB70CKC CHECK ((NO_W_ID BETWEEN
82801 AND 84000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB70 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB71.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB71 OFF;
ALTER TABLE NEW_ORDERB71 DROP CONSTRAINT
NEW_ORDERB71CKC;
ALTER TABLE NEW_ORDERB71 ADD CONSTRAINT
NEW_ORDERB71CKC CHECK ((NO_W_ID BETWEEN
84001 AND 85200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB71 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB72.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB72 OFF;
ALTER TABLE NEW_ORDERB72 DROP CONSTRAINT
NEW_ORDERB72CKC;
ALTER TABLE NEW_ORDERB72 ADD CONSTRAINT
NEW_ORDERB72CKC CHECK ((NO_W_ID BETWEEN
85201 AND 86400) AND (NO_O_ID >= 3676));
```

```
SET INTEGRITY FOR NEW_ORDERB72 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB73.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB73 OFF;
ALTER TABLE NEW_ORDERB73 DROP CONSTRAINT
NEW_ORDERB73CKC;
ALTER TABLE NEW_ORDERB73 ADD CONSTRAINT
NEW_ORDERB73CKC CHECK ((NO_W_ID BETWEEN
86401 AND 87600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB73 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB74.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB74 OFF;
ALTER TABLE NEW_ORDERB74 DROP CONSTRAINT
NEW_ORDERB74CKC;
ALTER TABLE NEW_ORDERB74 ADD CONSTRAINT
NEW_ORDERB74CKC CHECK ((NO_W_ID BETWEEN
87601 AND 88800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB74 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB75.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB75 OFF;
ALTER TABLE NEW_ORDERB75 DROP CONSTRAINT
NEW_ORDERB75CKC;
ALTER TABLE NEW_ORDERB75 ADD CONSTRAINT
NEW_ORDERB75CKC CHECK ((NO_W_ID BETWEEN
88801 AND 90000) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB75 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB76.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB76 OFF;
```

```
ALTER TABLE NEW_ORDERB76 DROP CONSTRAINT
NEW_ORDERB76CKC;
ALTER TABLE NEW_ORDERB76 ADD CONSTRAINT
NEW_ORDERB76CKC CHECK ((NO_W_ID BETWEEN
90001 AND 91200) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB76 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB77.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB77 OFF;
ALTER TABLE NEW_ORDERB77 DROP CONSTRAINT
NEW_ORDERB77CKC;
ALTER TABLE NEW_ORDERB77 ADD CONSTRAINT
NEW_ORDERB77CKC CHECK ((NO_W_ID BETWEEN
91201 AND 92400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB77 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB78.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB78 OFF;
ALTER TABLE NEW_ORDERB78 DROP CONSTRAINT
NEW_ORDERB78CKC;
ALTER TABLE NEW_ORDERB78 ADD CONSTRAINT
NEW_ORDERB78CKC CHECK ((NO_W_ID BETWEEN
92401 AND 93600) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB78 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB79.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB79 OFF;
ALTER TABLE NEW_ORDERB79 DROP CONSTRAINT
NEW_ORDERB79CKC;
ALTER TABLE NEW_ORDERB79 ADD CONSTRAINT
NEW_ORDERB79CKC CHECK ((NO_W_ID BETWEEN
93601 AND 94800) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB79 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 7201
AND 8400) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST NEW ORDERB80.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB80 OFF;
ALTER TABLE NEW_ORDERB80 DROP CONSTRAINT
NEW_ORDERB80CKC;
ALTER TABLE NEW_ORDERB80 ADD CONSTRAINT
NEW_ORDERB80CKC CHECK ((NO_W_ID >= 94801) AND
(NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDERB80 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 8401
AND 9600) 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 9601
AND 10800) 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 10801
AND 12000);
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 12001
AND 13200);
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 13201
AND 14400);
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 14401
AND 15600);

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 15601
AND 16800);
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 16801
AND 18000);
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 18001
AND 19200);
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 19201
AND 20400);
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 20401
AND 21600);
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 21601
AND 22800);
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
1200);
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 22801
AND 24000);
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 24001
AND 25200);
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 25201
AND 26400);
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 26401
AND 27600);
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 27601
AND 28800);
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 28801
AND 30000);
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 30001
AND 31200);
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 31201
AND 32400);
```

```
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 32401
AND 33600);
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 33601
AND 34800);
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 1201
AND 2400);
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 34801
AND 36000);
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 36001
AND 37200);
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 BETWEEN 37201
AND 38400);
SET INTEGRITY FOR ORDER_LINE32 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE33.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE33 OFF;
ALTER TABLE ORDER_LINE33 DROP CONSTRAINT
ORDER_LINE33CKC;
ALTER TABLE ORDER_LINE33 ADD CONSTRAINT
ORDER_LINE33CKC CHECK (OL_W_ID BETWEEN 38401
AND 39600);
SET INTEGRITY FOR ORDER_LINE33 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE34.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE34 OFF;
ALTER TABLE ORDER_LINE34 DROP CONSTRAINT
ORDER_LINE34CKC;
ALTER TABLE ORDER_LINE34 ADD CONSTRAINT
ORDER_LINE34CKC CHECK (OL_W_ID BETWEEN 39601
AND 40800);
SET INTEGRITY FOR ORDER_LINE34 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE35.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE35 OFF;
ALTER TABLE ORDER_LINE35 DROP CONSTRAINT
ORDER_LINE35CKC;
ALTER TABLE ORDER_LINE35 ADD CONSTRAINT
ORDER_LINE35CKC CHECK (OL_W_ID BETWEEN 40801
AND 42000);
SET INTEGRITY FOR ORDER_LINE35 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE36.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE36 OFF;
ALTER TABLE ORDER_LINE36 DROP CONSTRAINT
ORDER_LINE36CKC;
ALTER TABLE ORDER_LINE36 ADD CONSTRAINT
ORDER_LINE36CKC CHECK (OL_W_ID BETWEEN 42001
AND 43200);
SET INTEGRITY FOR ORDER_LINE36 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE37.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE37 OFF;
ALTER TABLE ORDER_LINE37 DROP CONSTRAINT
ORDER_LINE37CKC;
ALTER TABLE ORDER_LINE37 ADD CONSTRAINT
ORDER_LINE37CKC CHECK (OL_W_ID BETWEEN 43201
AND 44400);
SET INTEGRITY FOR ORDER_LINE37 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE38.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE38 OFF;
ALTER TABLE ORDER_LINE38 DROP CONSTRAINT
ORDER_LINE38CKC;
ALTER TABLE ORDER_LINE38 ADD CONSTRAINT
ORDER_LINE38CKC CHECK (OL_W_ID BETWEEN 44401
AND 45600);
SET INTEGRITY FOR ORDER_LINE38 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE39.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE39 OFF;
ALTER TABLE ORDER_LINE39 DROP CONSTRAINT
ORDER_LINE39CKC;
ALTER TABLE ORDER_LINE39 ADD CONSTRAINT
ORDER_LINE39CKC CHECK (OL_W_ID BETWEEN 45601
AND 46800);
SET INTEGRITY FOR ORDER_LINE39 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 2401
AND 3600);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE40.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE40 OFF;
ALTER TABLE ORDER_LINE40 DROP CONSTRAINT
ORDER_LINE40CKC;
ALTER TABLE ORDER_LINE40 ADD CONSTRAINT
ORDER_LINE40CKC CHECK (OL_W_ID BETWEEN 46801
AND 48000);
```

```
SET INTEGRITY FOR ORDER_LINE40 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE41.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE41 OFF;
ALTER TABLE ORDER_LINE41 DROP CONSTRAINT
ORDER_LINE41CKC;
ALTER TABLE ORDER_LINE41 ADD CONSTRAINT
ORDER_LINE41CKC CHECK (OL_W_ID BETWEEN 48001
AND 49200);
SET INTEGRITY FOR ORDER_LINE41 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE42.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE42 OFF;
ALTER TABLE ORDER_LINE42 DROP CONSTRAINT
ORDER_LINE42CKC;
ALTER TABLE ORDER_LINE42 ADD CONSTRAINT
ORDER_LINE42CKC CHECK (OL_W_ID BETWEEN 49201
AND 50400);
SET INTEGRITY FOR ORDER_LINE42 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE43.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE43 OFF;
ALTER TABLE ORDER_LINE43 DROP CONSTRAINT
ORDER_LINE43CKC;
ALTER TABLE ORDER_LINE43 ADD CONSTRAINT
ORDER_LINE43CKC CHECK (OL_W_ID BETWEEN 50401
AND 51600);
SET INTEGRITY FOR ORDER_LINE43 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE44.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE44 OFF;
```

```
ALTER TABLE ORDER_LINE44 DROP CONSTRAINT
ORDER_LINE44CKC;
ALTER TABLE ORDER_LINE44 ADD CONSTRAINT
ORDER_LINE44CKC CHECK (OL_W_ID BETWEEN 51601
AND 52800);
SET INTEGRITY FOR ORDER_LINE44 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE45.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE45 OFF;
ALTER TABLE ORDER_LINE45 DROP CONSTRAINT
ORDER_LINE45CKC;
ALTER TABLE ORDER_LINE45 ADD CONSTRAINT
ORDER_LINE45CKC CHECK (OL_W_ID BETWEEN 52801
AND 54000);
SET INTEGRITY FOR ORDER_LINE45 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE46.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE46 OFF;
ALTER TABLE ORDER_LINE46 DROP CONSTRAINT
ORDER_LINE46CKC;
ALTER TABLE ORDER_LINE46 ADD CONSTRAINT
ORDER_LINE46CKC CHECK (OL_W_ID BETWEEN 54001
AND 55200);
SET INTEGRITY FOR ORDER_LINE46 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE47.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE47 OFF;
ALTER TABLE ORDER_LINE47 DROP CONSTRAINT
ORDER_LINE47CKC;
ALTER TABLE ORDER_LINE47 ADD CONSTRAINT
ORDER_LINE47CKC CHECK (OL_W_ID BETWEEN 55201
AND 56400);
SET INTEGRITY FOR ORDER_LINE47 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE48.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE48 OFF;
ALTER TABLE ORDER_LINE48 DROP CONSTRAINT
ORDER_LINE48CKC;
ALTER TABLE ORDER_LINE48 ADD CONSTRAINT
ORDER_LINE48CKC CHECK (OL_W_ID BETWEEN 56401
AND 57600);
SET INTEGRITY FOR ORDER_LINE48 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE49.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE49 OFF;
ALTER TABLE ORDER_LINE49 DROP CONSTRAINT
ORDER_LINE49CKC;
ALTER TABLE ORDER_LINE49 ADD CONSTRAINT
ORDER_LINE49CKC CHECK (OL_W_ID BETWEEN 57601
AND 58800);
SET INTEGRITY FOR ORDER_LINE49 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 3601
AND 4800);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE50.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE50 OFF;
ALTER TABLE ORDER_LINE50 DROP CONSTRAINT
ORDER_LINE50CKC;
ALTER TABLE ORDER_LINE50 ADD CONSTRAINT
ORDER_LINE50CKC CHECK (OL_W_ID BETWEEN 58801
AND 60000);
SET INTEGRITY FOR ORDER_LINE50 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE51.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE51 OFF;
ALTER TABLE ORDER_LINE51 DROP CONSTRAINT
ORDER_LINE51CKC;
ALTER TABLE ORDER_LINE51 ADD CONSTRAINT
ORDER_LINE51CKC CHECK (OL_W_ID BETWEEN 60001
AND 61200);
SET INTEGRITY FOR ORDER_LINE51 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE52.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE52 OFF;
ALTER TABLE ORDER_LINE52 DROP CONSTRAINT
ORDER_LINE52CKC;
ALTER TABLE ORDER_LINE52 ADD CONSTRAINT
ORDER_LINE52CKC CHECK (OL_W_ID BETWEEN 61201
AND 62400);
SET INTEGRITY FOR ORDER_LINE52 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE53.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE53 OFF;
ALTER TABLE ORDER_LINE53 DROP CONSTRAINT
ORDER_LINE53CKC;
ALTER TABLE ORDER_LINE53 ADD CONSTRAINT
ORDER_LINE53CKC CHECK (OL_W_ID BETWEEN 62401
AND 63600);
SET INTEGRITY FOR ORDER_LINE53 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE54.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE54 OFF;
ALTER TABLE ORDER_LINE54 DROP CONSTRAINT
ORDER_LINE54CKC;
ALTER TABLE ORDER_LINE54 ADD CONSTRAINT
ORDER_LINE54CKC CHECK (OL_W_ID BETWEEN 63601
AND 64800);

SET INTEGRITY FOR ORDER_LINE54 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE55.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE55 OFF;
ALTER TABLE ORDER_LINE55 DROP CONSTRAINT
ORDER_LINE55CKC;
ALTER TABLE ORDER_LINE55 ADD CONSTRAINT
ORDER_LINE55CKC CHECK (OL_W_ID BETWEEN 64801
AND 66000);
SET INTEGRITY FOR ORDER_LINE55 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE56.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE56 OFF;
ALTER TABLE ORDER_LINE56 DROP CONSTRAINT
ORDER_LINE56CKC;
ALTER TABLE ORDER_LINE56 ADD CONSTRAINT
ORDER_LINE56CKC CHECK (OL_W_ID BETWEEN 66001
AND 67200);
SET INTEGRITY FOR ORDER_LINE56 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE57.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE57 OFF;
ALTER TABLE ORDER_LINE57 DROP CONSTRAINT
ORDER_LINE57CKC;
ALTER TABLE ORDER_LINE57 ADD CONSTRAINT
ORDER_LINE57CKC CHECK (OL_W_ID BETWEEN 67201
AND 68400);
SET INTEGRITY FOR ORDER_LINE57 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE58.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE58 OFF;

ALTER TABLE ORDER_LINE58 DROP CONSTRAINT
ORDER_LINE58CKC;
ALTER TABLE ORDER_LINE58 ADD CONSTRAINT
ORDER_LINE58CKC CHECK (OL_W_ID BETWEEN 68401
AND 69600);
SET INTEGRITY FOR ORDER_LINE58 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE59.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE59 OFF;
ALTER TABLE ORDER_LINE59 DROP CONSTRAINT
ORDER_LINE59CKC;
ALTER TABLE ORDER_LINE59 ADD CONSTRAINT
ORDER_LINE59CKC CHECK (OL_W_ID BETWEEN 69601
AND 70800);
SET INTEGRITY FOR ORDER_LINE59 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 4801
AND 6000);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE60.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE60 OFF;
ALTER TABLE ORDER_LINE60 DROP CONSTRAINT
ORDER_LINE60CKC;
ALTER TABLE ORDER_LINE60 ADD CONSTRAINT
ORDER_LINE60CKC CHECK (OL_W_ID BETWEEN 70801
AND 72000);
SET INTEGRITY FOR ORDER_LINE60 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE61.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE61 OFF;
ALTER TABLE ORDER_LINE61 DROP CONSTRAINT
ORDER_LINE61CKC;
ALTER TABLE ORDER_LINE61 ADD CONSTRAINT
ORDER_LINE61CKC CHECK (OL_W_ID BETWEEN 72001
AND 73200);
SET INTEGRITY FOR ORDER_LINE61 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE62.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE62 OFF;
ALTER TABLE ORDER_LINE62 DROP CONSTRAINT
ORDER_LINE62CKC;
ALTER TABLE ORDER_LINE62 ADD CONSTRAINT
ORDER_LINE62CKC CHECK (OL_W_ID BETWEEN 73201
AND 74400);
SET INTEGRITY FOR ORDER_LINE62 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE63.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE63 OFF;
ALTER TABLE ORDER_LINE63 DROP CONSTRAINT
ORDER_LINE63CKC;
ALTER TABLE ORDER_LINE63 ADD CONSTRAINT
ORDER_LINE63CKC CHECK (OL_W_ID BETWEEN 74401
AND 75600);
SET INTEGRITY FOR ORDER_LINE63 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE64.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE64 OFF;
ALTER TABLE ORDER_LINE64 DROP CONSTRAINT
ORDER_LINE64CKC;
ALTER TABLE ORDER_LINE64 ADD CONSTRAINT
ORDER_LINE64CKC CHECK (OL_W_ID BETWEEN 75601
AND 76800);
SET INTEGRITY FOR ORDER_LINE64 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE65.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE65 OFF;
ALTER TABLE ORDER_LINE65 DROP CONSTRAINT
ORDER_LINE65CKC;
ALTER TABLE ORDER_LINE65 ADD CONSTRAINT
ORDER_LINE65CKC CHECK (OL_W_ID BETWEEN 76801
AND 78000);
SET INTEGRITY FOR ORDER_LINE65 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE66.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE66 OFF;
ALTER TABLE ORDER_LINE66 DROP CONSTRAINT
ORDER_LINE66CKC;
ALTER TABLE ORDER_LINE66 ADD CONSTRAINT
ORDER_LINE66CKC CHECK (OL_W_ID BETWEEN 78001
AND 79200);
SET INTEGRITY FOR ORDER_LINE66 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE67.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE67 OFF;
ALTER TABLE ORDER_LINE67 DROP CONSTRAINT
ORDER_LINE67CKC;
ALTER TABLE ORDER_LINE67 ADD CONSTRAINT
ORDER_LINE67CKC CHECK (OL_W_ID BETWEEN 79201
AND 80400);
SET INTEGRITY FOR ORDER_LINE67 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE68.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE68 OFF;
ALTER TABLE ORDER_LINE68 DROP CONSTRAINT
ORDER_LINE68CKC;
ALTER TABLE ORDER_LINE68 ADD CONSTRAINT
ORDER_LINE68CKC CHECK (OL_W_ID BETWEEN 80401
AND 81600);

SET INTEGRITY FOR ORDER_LINE68 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE69.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE69 OFF;
ALTER TABLE ORDER_LINE69 DROP CONSTRAINT
ORDER_LINE69CKC;
ALTER TABLE ORDER_LINE69 ADD CONSTRAINT
ORDER_LINE69CKC CHECK (OL_W_ID BETWEEN 81601
AND 82800);
SET INTEGRITY FOR ORDER_LINE69 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 6001
AND 7200);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE70.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE70 OFF;
ALTER TABLE ORDER_LINE70 DROP CONSTRAINT
ORDER_LINE70CKC;
ALTER TABLE ORDER_LINE70 ADD CONSTRAINT
ORDER_LINE70CKC CHECK (OL_W_ID BETWEEN 82801
AND 84000);
SET INTEGRITY FOR ORDER_LINE70 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDER LINE71.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE71 OFF;

```
ALTER TABLE ORDER_LINE71 DROP CONSTRAINT
ORDER_LINE71CKC;
ALTER TABLE ORDER_LINE71 ADD CONSTRAINT
ORDER_LINE71CKC CHECK (OL_W_ID BETWEEN 84001
AND 85200);
SET INTEGRITY FOR ORDER_LINE71 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE72.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE72 OFF;
ALTER TABLE ORDER_LINE72 DROP CONSTRAINT
ORDER_LINE72CKC;
ALTER TABLE ORDER_LINE72 ADD CONSTRAINT
ORDER_LINE72CKC CHECK (OL_W_ID BETWEEN 85201
AND 86400);
SET INTEGRITY FOR ORDER_LINE72 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE73.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE73 OFF;
ALTER TABLE ORDER_LINE73 DROP CONSTRAINT
ORDER_LINE73CKC;
ALTER TABLE ORDER_LINE73 ADD CONSTRAINT
ORDER_LINE73CKC CHECK (OL_W_ID BETWEEN 86401
AND 87600);
SET INTEGRITY FOR ORDER_LINE73 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE74.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE74 OFF;
ALTER TABLE ORDER_LINE74 DROP CONSTRAINT
ORDER_LINE74CKC;
ALTER TABLE ORDER_LINE74 ADD CONSTRAINT
ORDER_LINE74CKC CHECK (OL_W_ID BETWEEN 87601
AND 88800);
SET INTEGRITY FOR ORDER_LINE74 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE75.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE75 OFF;
ALTER TABLE ORDER_LINE75 DROP CONSTRAINT
ORDER_LINE75CKC;
ALTER TABLE ORDER_LINE75 ADD CONSTRAINT
ORDER_LINE75CKC CHECK (OL_W_ID BETWEEN 88801
AND 90000);
SET INTEGRITY FOR ORDER_LINE75 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE76.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE76 OFF;
ALTER TABLE ORDER_LINE76 DROP CONSTRAINT
ORDER_LINE76CKC;
ALTER TABLE ORDER_LINE76 ADD CONSTRAINT
ORDER_LINE76CKC CHECK (OL_W_ID BETWEEN 90001
AND 91200);
SET INTEGRITY FOR ORDER_LINE76 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE77.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE77 OFF;
ALTER TABLE ORDER_LINE77 DROP CONSTRAINT
ORDER_LINE77CKC;
ALTER TABLE ORDER_LINE77 ADD CONSTRAINT
ORDER_LINE77CKC CHECK (OL_W_ID BETWEEN 91201
AND 92400);
SET INTEGRITY FOR ORDER_LINE77 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE78.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE78 OFF;
ALTER TABLE ORDER_LINE78 DROP CONSTRAINT
ORDER_LINE78CKC;
ALTER TABLE ORDER_LINE78 ADD CONSTRAINT
ORDER_LINE78CKC CHECK (OL_W_ID BETWEEN 92401
AND 93600);
SET INTEGRITY FOR ORDER_LINE78 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE79.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE79 OFF;
ALTER TABLE ORDER_LINE79 DROP CONSTRAINT
ORDER_LINE79CKC;
ALTER TABLE ORDER_LINE79 ADD CONSTRAINT
ORDER_LINE79CKC CHECK (OL_W_ID BETWEEN 93601
AND 94800);
SET INTEGRITY FOR ORDER_LINE79 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 7201
AND 8400);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDER LINE80.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE80 OFF;
ALTER TABLE ORDER_LINE80 DROP CONSTRAINT
ORDER_LINE80CKC;
ALTER TABLE ORDER_LINE80 ADD CONSTRAINT
ORDER_LINE80CKC CHECK (OL_W_ID >= 94801);
SET INTEGRITY FOR ORDER_LINE80 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 8401
AND 9600);
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 9601
AND 10800);
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 10801 AND
12000);
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 12001 AND
13200);
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 13201 AND
14400);
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 14401 AND
15600);
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 15601 AND
16800);
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 16801 AND
18000);
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 18001 AND
19200);
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 19201 AND
20400);
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 20401 AND
21600);
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 21601 AND
22800);
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 1200);
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 22801 AND
24000);
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 24001 AND
25200);
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 25201 AND
26400);
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 26401 AND
27600);
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 27601 AND
28800);
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 28801 AND
30000);
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 30001 AND
31200);
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 31201 AND
32400);
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 32401 AND
33600);
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 33601 AND
34800);
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 1201 AND
2400);
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 34801 AND
36000);
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 36001 AND
37200);
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 BETWEEN 37201 AND
38400);
SET INTEGRITY FOR ORDERS32 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS33.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS33 OFF;
ALTER TABLE ORDERS33 DROP CONSTRAINT
ORDERS33CKC;
ALTER TABLE ORDERS33 ADD CONSTRAINT
ORDERS33CKC CHECK (O_W_ID BETWEEN 38401 AND
39600);
SET INTEGRITY FOR ORDERS33 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS34.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS34 OFF;
ALTER TABLE ORDERS34 DROP CONSTRAINT
ORDERS34CKC;
ALTER TABLE ORDERS34 ADD CONSTRAINT
ORDERS34CKC CHECK (O_W_ID BETWEEN 39601 AND
40800);
SET INTEGRITY FOR ORDERS34 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS35.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS35 OFF;
ALTER TABLE ORDERS35 DROP CONSTRAINT
ORDERS35CKC;
ALTER TABLE ORDERS35 ADD CONSTRAINT
ORDERS35CKC CHECK (O_W_ID BETWEEN 40801 AND
42000);
SET INTEGRITY FOR ORDERS35 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS36.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS36 OFF;
ALTER TABLE ORDERS36 DROP CONSTRAINT
ORDERS36CKC;
ALTER TABLE ORDERS36 ADD CONSTRAINT
ORDERS36CKC CHECK (O_W_ID BETWEEN 42001 AND
43200);
```

```
SET INTEGRITY FOR ORDERS36 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS37.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS37 OFF;
ALTER TABLE ORDERS37 DROP CONSTRAINT
ORDERS37CKC;
ALTER TABLE ORDERS37 ADD CONSTRAINT
ORDERS37CKC CHECK (O_W_ID BETWEEN 43201 AND
44400);
SET INTEGRITY FOR ORDERS37 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS38.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS38 OFF;
ALTER TABLE ORDERS38 DROP CONSTRAINT
ORDERS38CKC;
ALTER TABLE ORDERS38 ADD CONSTRAINT
ORDERS38CKC CHECK (O_W_ID BETWEEN 44401 AND
45600);
SET INTEGRITY FOR ORDERS38 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS39.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS39 OFF;
ALTER TABLE ORDERS39 DROP CONSTRAINT
ORDERS39CKC;
ALTER TABLE ORDERS39 ADD CONSTRAINT
ORDERS39CKC CHECK (O_W_ID BETWEEN 45601 AND
46800);
SET INTEGRITY FOR ORDERS39 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 2401 AND
3600);
SET INTEGRITY FOR ORDERS3 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS40.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS40 OFF;
ALTER TABLE ORDERS40 DROP CONSTRAINT
ORDERS40CKC;
ALTER TABLE ORDERS40 ADD CONSTRAINT
ORDERS40CKC CHECK (O_W_ID BETWEEN 46801 AND
48000);
SET INTEGRITY FOR ORDERS40 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS41.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS41 OFF;
ALTER TABLE ORDERS41 DROP CONSTRAINT
ORDERS41CKC;
ALTER TABLE ORDERS41 ADD CONSTRAINT
ORDERS41CKC CHECK (O_W_ID BETWEEN 48001 AND
49200);
SET INTEGRITY FOR ORDERS41 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS42.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS42 OFF;
ALTER TABLE ORDERS42 DROP CONSTRAINT
ORDERS42CKC;
ALTER TABLE ORDERS42 ADD CONSTRAINT
ORDERS42CKC CHECK (O_W_ID BETWEEN 49201 AND
50400);
SET INTEGRITY FOR ORDERS42 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS43.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS43 OFF;
ALTER TABLE ORDERS43 DROP CONSTRAINT
ORDERS43CKC;
ALTER TABLE ORDERS43 ADD CONSTRAINT
ORDERS43CKC CHECK (O_W_ID BETWEEN 50401 AND
51600);
SET INTEGRITY FOR ORDERS43 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS44.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS44 OFF;
ALTER TABLE ORDERS44 DROP CONSTRAINT
ORDERS44CKC;
ALTER TABLE ORDERS44 ADD CONSTRAINT
ORDERS44CKC CHECK (O_W_ID BETWEEN 51601 AND
52800);
SET INTEGRITY FOR ORDERS44 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS45.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS45 OFF;
ALTER TABLE ORDERS45 DROP CONSTRAINT
ORDERS45CKC;
ALTER TABLE ORDERS45 ADD CONSTRAINT
ORDERS45CKC CHECK (O_W_ID BETWEEN 52801 AND
54000);
SET INTEGRITY FOR ORDERS45 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS46.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS46 OFF;
ALTER TABLE ORDERS46 DROP CONSTRAINT
ORDERS46CKC;
ALTER TABLE ORDERS46 ADD CONSTRAINT
ORDERS46CKC CHECK (O_W_ID BETWEEN 54001 AND
55200);
SET INTEGRITY FOR ORDERS46 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS47.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS47 OFF;
ALTER TABLE ORDERS47 DROP CONSTRAINT
ORDERS47CKC;
ALTER TABLE ORDERS47 ADD CONSTRAINT
ORDERS47CKC CHECK (O_W_ID BETWEEN 55201 AND
56400);
SET INTEGRITY FOR ORDERS47 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS48.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS48 OFF;
ALTER TABLE ORDERS48 DROP CONSTRAINT
ORDERS48CKC;
ALTER TABLE ORDERS48 ADD CONSTRAINT
ORDERS48CKC CHECK (O_W_ID BETWEEN 56401 AND
57600);
SET INTEGRITY FOR ORDERS48 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST ORDERS49.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS49 OFF;
ALTER TABLE ORDERS49 DROP CONSTRAINT
ORDERS49CKC;
ALTER TABLE ORDERS49 ADD CONSTRAINT
ORDERS49CKC CHECK (O_W_ID BETWEEN 57601 AND
58800);
SET INTEGRITY FOR ORDERS49 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 3601 AND
4800);
```

SET INTEGRITY FOR ORDERS4 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS50.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS50 OFF;
ALTER TABLE ORDERS50 DROP CONSTRAINT
ORDERS50CKC;
ALTER TABLE ORDERS50 ADD CONSTRAINT
ORDERS50CKC CHECK (O_W_ID BETWEEN 58801 AND
60000);
SET INTEGRITY FOR ORDERS50 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS51.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS51 OFF;
ALTER TABLE ORDERS51 DROP CONSTRAINT
ORDERS51CKC;
ALTER TABLE ORDERS51 ADD CONSTRAINT
ORDERS51CKC CHECK (O_W_ID BETWEEN 60001 AND
61200);
SET INTEGRITY FOR ORDERS51 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS52.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS52 OFF;
ALTER TABLE ORDERS52 DROP CONSTRAINT
ORDERS52CKC;
ALTER TABLE ORDERS52 ADD CONSTRAINT
ORDERS52CKC CHECK (O_W_ID BETWEEN 61201 AND
62400);
SET INTEGRITY FOR ORDERS52 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS53.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS53 OFF;

ALTER TABLE ORDERS53 DROP CONSTRAINT
ORDERS53CKC;
ALTER TABLE ORDERS53 ADD CONSTRAINT
ORDERS53CKC CHECK (O_W_ID BETWEEN 62401 AND
63600);
SET INTEGRITY FOR ORDERS53 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS54.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS54 OFF;
ALTER TABLE ORDERS54 DROP CONSTRAINT
ORDERS54CKC;
ALTER TABLE ORDERS54 ADD CONSTRAINT
ORDERS54CKC CHECK (O_W_ID BETWEEN 63601 AND
64800);
SET INTEGRITY FOR ORDERS54 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS55.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS55 OFF;
ALTER TABLE ORDERS55 DROP CONSTRAINT
ORDERS55CKC;
ALTER TABLE ORDERS55 ADD CONSTRAINT
ORDERS55CKC CHECK (O_W_ID BETWEEN 64801 AND
66000);
SET INTEGRITY FOR ORDERS55 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS56.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS56 OFF;
ALTER TABLE ORDERS56 DROP CONSTRAINT
ORDERS56CKC;
ALTER TABLE ORDERS56 ADD CONSTRAINT
ORDERS56CKC CHECK (O_W_ID BETWEEN 66001 AND
67200);
SET INTEGRITY FOR ORDERS56 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS57.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS57 OFF;
ALTER TABLE ORDERS57 DROP CONSTRAINT
ORDERS57CKC;
ALTER TABLE ORDERS57 ADD CONSTRAINT
ORDERS57CKC CHECK (O_W_ID BETWEEN 67201 AND
68400);
SET INTEGRITY FOR ORDERS57 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS58.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS58 OFF;
ALTER TABLE ORDERS58 DROP CONSTRAINT
ORDERS58CKC;
ALTER TABLE ORDERS58 ADD CONSTRAINT
ORDERS58CKC CHECK (O_W_ID BETWEEN 68401 AND
69600);
SET INTEGRITY FOR ORDERS58 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS59.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS59 OFF;
ALTER TABLE ORDERS59 DROP CONSTRAINT
ORDERS59CKC;
ALTER TABLE ORDERS59 ADD CONSTRAINT
ORDERS59CKC CHECK (O_W_ID BETWEEN 69601 AND
70800);
SET INTEGRITY FOR ORDERS59 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 4801 AND
6000);
SET INTEGRITY FOR ORDERS5 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS60.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS60 OFF;
ALTER TABLE ORDERS60 DROP CONSTRAINT
ORDERS60CKC;
ALTER TABLE ORDERS60 ADD CONSTRAINT
ORDERS60CKC CHECK (O_W_ID BETWEEN 70801 AND
72000);
SET INTEGRITY FOR ORDERS60 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS61.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS61 OFF;
ALTER TABLE ORDERS61 DROP CONSTRAINT
ORDERS61CKC;
ALTER TABLE ORDERS61 ADD CONSTRAINT
ORDERS61CKC CHECK (O_W_ID BETWEEN 72001 AND
73200);
SET INTEGRITY FOR ORDERS61 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS62.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS62 OFF;
ALTER TABLE ORDERS62 DROP CONSTRAINT
ORDERS62CKC;
ALTER TABLE ORDERS62 ADD CONSTRAINT
ORDERS62CKC CHECK (O_W_ID BETWEEN 73201 AND
74400);
SET INTEGRITY FOR ORDERS62 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS63.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS63 OFF;
ALTER TABLE ORDERS63 DROP CONSTRAINT
ORDERS63CKC;
ALTER TABLE ORDERS63 ADD CONSTRAINT
ORDERS63CKC CHECK (O_W_ID BETWEEN 74401 AND
75600);

SET INTEGRITY FOR ORDERS63 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS64.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS64 OFF;
ALTER TABLE ORDERS64 DROP CONSTRAINT
ORDERS64CKC;
ALTER TABLE ORDERS64 ADD CONSTRAINT
ORDERS64CKC CHECK (O_W_ID BETWEEN 75601 AND
76800);
SET INTEGRITY FOR ORDERS64 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS65.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS65 OFF;
ALTER TABLE ORDERS65 DROP CONSTRAINT
ORDERS65CKC;
ALTER TABLE ORDERS65 ADD CONSTRAINT
ORDERS65CKC CHECK (O_W_ID BETWEEN 76801 AND
78000);
SET INTEGRITY FOR ORDERS65 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS66.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS66 OFF;
ALTER TABLE ORDERS66 DROP CONSTRAINT
ORDERS66CKC;
ALTER TABLE ORDERS66 ADD CONSTRAINT
ORDERS66CKC CHECK (O_W_ID BETWEEN 78001 AND
79200);
SET INTEGRITY FOR ORDERS66 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS67.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS67 OFF;

ALTER TABLE ORDERS67 DROP CONSTRAINT
ORDERS67CKC;
ALTER TABLE ORDERS67 ADD CONSTRAINT
ORDERS67CKC CHECK (O_W_ID BETWEEN 79201 AND
80400);
SET INTEGRITY FOR ORDERS67 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS68.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS68 OFF;
ALTER TABLE ORDERS68 DROP CONSTRAINT
ORDERS68CKC;
ALTER TABLE ORDERS68 ADD CONSTRAINT
ORDERS68CKC CHECK (O_W_ID BETWEEN 80401 AND
81600);
SET INTEGRITY FOR ORDERS68 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS69.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS69 OFF;
ALTER TABLE ORDERS69 DROP CONSTRAINT
ORDERS69CKC;
ALTER TABLE ORDERS69 ADD CONSTRAINT
ORDERS69CKC CHECK (O_W_ID BETWEEN 81601 AND
82800);
SET INTEGRITY FOR ORDERS69 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 6001 AND
7200);
SET INTEGRITY FOR ORDERS6 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS70.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS70 OFF;
ALTER TABLE ORDERS70 DROP CONSTRAINT
ORDERS70CKC;
ALTER TABLE ORDERS70 ADD CONSTRAINT
ORDERS70CKC CHECK (O_W_ID BETWEEN 82801 AND
84000);
SET INTEGRITY FOR ORDERS70 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS71.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS71 OFF;
ALTER TABLE ORDERS71 DROP CONSTRAINT
ORDERS71CKC;
ALTER TABLE ORDERS71 ADD CONSTRAINT
ORDERS71CKC CHECK (O_W_ID BETWEEN 84001 AND
85200);
SET INTEGRITY FOR ORDERS71 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS72.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS72 OFF;
ALTER TABLE ORDERS72 DROP CONSTRAINT
ORDERS72CKC;
ALTER TABLE ORDERS72 ADD CONSTRAINT
ORDERS72CKC CHECK (O_W_ID BETWEEN 85201 AND
86400);
SET INTEGRITY FOR ORDERS72 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS73.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS73 OFF;
ALTER TABLE ORDERS73 DROP CONSTRAINT
ORDERS73CKC;
ALTER TABLE ORDERS73 ADD CONSTRAINT
ORDERS73CKC CHECK (O_W_ID BETWEEN 86401 AND
87600);
SET INTEGRITY FOR ORDERS73 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS74.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS74 OFF;
ALTER TABLE ORDERS74 DROP CONSTRAINT
ORDERS74CKC;
ALTER TABLE ORDERS74 ADD CONSTRAINT
ORDERS74CKC CHECK (O_W_ID BETWEEN 87601 AND
88800);
SET INTEGRITY FOR ORDERS74 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS75.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS75 OFF;
ALTER TABLE ORDERS75 DROP CONSTRAINT
ORDERS75CKC;
ALTER TABLE ORDERS75 ADD CONSTRAINT
ORDERS75CKC CHECK (O_W_ID BETWEEN 88801 AND
90000);
SET INTEGRITY FOR ORDERS75 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS76.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS76 OFF;
ALTER TABLE ORDERS76 DROP CONSTRAINT
ORDERS76CKC;
ALTER TABLE ORDERS76 ADD CONSTRAINT
ORDERS76CKC CHECK (O_W_ID BETWEEN 90001 AND
91200);
SET INTEGRITY FOR ORDERS76 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS77.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS77 OFF;
ALTER TABLE ORDERS77 DROP CONSTRAINT
ORDERS77CKC;
ALTER TABLE ORDERS77 ADD CONSTRAINT
ORDERS77CKC CHECK (O_W_ID BETWEEN 91201 AND
92400);

SET INTEGRITY FOR ORDERS77 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS78.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS78 OFF;
ALTER TABLE ORDERS78 DROP CONSTRAINT
ORDERS78CKC;
ALTER TABLE ORDERS78 ADD CONSTRAINT
ORDERS78CKC CHECK (O_W_ID BETWEEN 92401 AND
93600);
SET INTEGRITY FOR ORDERS78 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS79.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS79 OFF;
ALTER TABLE ORDERS79 DROP CONSTRAINT
ORDERS79CKC;
ALTER TABLE ORDERS79 ADD CONSTRAINT
ORDERS79CKC CHECK (O_W_ID BETWEEN 93601 AND
94800);
SET INTEGRITY FOR ORDERS79 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 7201 AND
8400);
SET INTEGRITY FOR ORDERS7 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST ORDERS80.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS80 OFF;

```
ALTER TABLE ORDERS80 DROP CONSTRAINT
ORDERS80CKC;
ALTER TABLE ORDERS80 ADD CONSTRAINT
ORDERS80CKC CHECK (O_W_ID >= 94801);
SET INTEGRITY FOR ORDERS80 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 8401 AND
9600);
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 9601 AND
10800);
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 10801 AND
12000);
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 12001 AND
13200);
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 13201 AND
14400);
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 14401 AND
15600);
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 15601 AND
16800);
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 16801 AND
18000);
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 18001 AND
19200);
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 19201 AND
20400);
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 20401 AND
21600);
```

```
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 21601 AND
22800);
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 1200);
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 22801 AND
24000);
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 24001 AND
25200);
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 25201 AND
26400);
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 26401 AND
27600);
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 27601 AND
28800);
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 28801 AND
30000);
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 30001 AND
31200);
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 31201 AND
32400);
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 32401 AND
33600);
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 33601 AND
34800);
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 1201 AND 2400);
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 34801 AND
36000);
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 36001 AND
37200);
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 BETWEEN 37201 AND
38400);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK33.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK33 OFF;
ALTER TABLE STOCK33 DROP CONSTRAINT
STOCK33CKC;
ALTER TABLE STOCK33 ADD CONSTRAINT
STOCK33CKC CHECK (S_W_ID BETWEEN 38401 AND
39600);
SET INTEGRITY FOR STOCK33 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK34.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK34 OFF;
ALTER TABLE STOCK34 DROP CONSTRAINT
STOCK34CKC;
ALTER TABLE STOCK34 ADD CONSTRAINT
STOCK34CKC CHECK (S_W_ID BETWEEN 39601 AND
40800);
SET INTEGRITY FOR STOCK34 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK35.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK35 OFF;
ALTER TABLE STOCK35 DROP CONSTRAINT
STOCK35CKC;
ALTER TABLE STOCK35 ADD CONSTRAINT
STOCK35CKC CHECK (S_W_ID BETWEEN 40801 AND
42000);

SET INTEGRITY FOR STOCK35 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK36.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK36 OFF;
ALTER TABLE STOCK36 DROP CONSTRAINT
STOCK36CKC;
ALTER TABLE STOCK36 ADD CONSTRAINT
STOCK36CKC CHECK (S_W_ID BETWEEN 42001 AND
43200);
SET INTEGRITY FOR STOCK36 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK37.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK37 OFF;
ALTER TABLE STOCK37 DROP CONSTRAINT
STOCK37CKC;
ALTER TABLE STOCK37 ADD CONSTRAINT
STOCK37CKC CHECK (S_W_ID BETWEEN 43201 AND
44400);
SET INTEGRITY FOR STOCK37 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK38.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK38 OFF;
ALTER TABLE STOCK38 DROP CONSTRAINT
STOCK38CKC;
ALTER TABLE STOCK38 ADD CONSTRAINT
STOCK38CKC CHECK (S_W_ID BETWEEN 44401 AND
45600);
SET INTEGRITY FOR STOCK38 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK39.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK39 OFF;

```
ALTER TABLE STOCK39 DROP CONSTRAINT
STOCK39CKC;
ALTER TABLE STOCK39 ADD CONSTRAINT
STOCK39CKC CHECK (S_W_ID BETWEEN 45601 AND
46800);
SET INTEGRITY FOR STOCK39 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 2401 AND 3600);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK40.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK40 OFF;
ALTER TABLE STOCK40 DROP CONSTRAINT
STOCK40CKC;
ALTER TABLE STOCK40 ADD CONSTRAINT
STOCK40CKC CHECK (S_W_ID BETWEEN 46801 AND
48000);
SET INTEGRITY FOR STOCK40 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK41.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK41 OFF;
ALTER TABLE STOCK41 DROP CONSTRAINT
STOCK41CKC;
ALTER TABLE STOCK41 ADD CONSTRAINT
STOCK41CKC CHECK (S_W_ID BETWEEN 48001 AND
49200);
SET INTEGRITY FOR STOCK41 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK42.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK42 OFF;
ALTER TABLE STOCK42 DROP CONSTRAINT
STOCK42CKC;
ALTER TABLE STOCK42 ADD CONSTRAINT
STOCK42CKC CHECK (S_W_ID BETWEEN 49201 AND
50400);
SET INTEGRITY FOR STOCK42 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK43.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK43 OFF;
ALTER TABLE STOCK43 DROP CONSTRAINT
STOCK43CKC;
ALTER TABLE STOCK43 ADD CONSTRAINT
STOCK43CKC CHECK (S_W_ID BETWEEN 50401 AND
51600);
SET INTEGRITY FOR STOCK43 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK44.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK44 OFF;
ALTER TABLE STOCK44 DROP CONSTRAINT
STOCK44CKC;
ALTER TABLE STOCK44 ADD CONSTRAINT
STOCK44CKC CHECK (S_W_ID BETWEEN 51601 AND
52800);
SET INTEGRITY FOR STOCK44 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK45.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK45 OFF;
ALTER TABLE STOCK45 DROP CONSTRAINT
STOCK45CKC;
ALTER TABLE STOCK45 ADD CONSTRAINT
STOCK45CKC CHECK (S_W_ID BETWEEN 52801 AND
54000);
SET INTEGRITY FOR STOCK45 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK46.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK46 OFF;
ALTER TABLE STOCK46 DROP CONSTRAINT
STOCK46CKC;
ALTER TABLE STOCK46 ADD CONSTRAINT
STOCK46CKC CHECK (S_W_ID BETWEEN 54001 AND
55200);
SET INTEGRITY FOR STOCK46 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK47.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK47 OFF;
ALTER TABLE STOCK47 DROP CONSTRAINT
STOCK47CKC;
ALTER TABLE STOCK47 ADD CONSTRAINT
STOCK47CKC CHECK (S_W_ID BETWEEN 55201 AND
56400);
SET INTEGRITY FOR STOCK47 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK48.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK48 OFF;
ALTER TABLE STOCK48 DROP CONSTRAINT
STOCK48CKC;
ALTER TABLE STOCK48 ADD CONSTRAINT
STOCK48CKC CHECK (S_W_ID BETWEEN 56401 AND
57600);
SET INTEGRITY FOR STOCK48 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK49.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK49 OFF;
ALTER TABLE STOCK49 DROP CONSTRAINT
STOCK49CKC;
ALTER TABLE STOCK49 ADD CONSTRAINT
STOCK49CKC CHECK (S_W_ID BETWEEN 57601 AND
58800);
```

```
SET INTEGRITY FOR STOCK49 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 3601 AND 4800);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK50.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK50 OFF;
ALTER TABLE STOCK50 DROP CONSTRAINT
STOCK50CKC;
ALTER TABLE STOCK50 ADD CONSTRAINT
STOCK50CKC CHECK (S_W_ID BETWEEN 58801 AND
60000);
SET INTEGRITY FOR STOCK50 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK51.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK51 OFF;
ALTER TABLE STOCK51 DROP CONSTRAINT
STOCK51CKC;
ALTER TABLE STOCK51 ADD CONSTRAINT
STOCK51CKC CHECK (S_W_ID BETWEEN 60001 AND
61200);
SET INTEGRITY FOR STOCK51 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK52.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK52 OFF;
ALTER TABLE STOCK52 DROP CONSTRAINT
STOCK52CKC;
```

```
ALTER TABLE STOCK52 ADD CONSTRAINT
STOCK52CKC CHECK (S_W_ID BETWEEN 61201 AND
62400);
SET INTEGRITY FOR STOCK52 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK53.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK53 OFF;
ALTER TABLE STOCK53 DROP CONSTRAINT
STOCK53CKC;
ALTER TABLE STOCK53 ADD CONSTRAINT
STOCK53CKC CHECK (S_W_ID BETWEEN 62401 AND
63600);
SET INTEGRITY FOR STOCK53 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK54.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK54 OFF;
ALTER TABLE STOCK54 DROP CONSTRAINT
STOCK54CKC;
ALTER TABLE STOCK54 ADD CONSTRAINT
STOCK54CKC CHECK (S_W_ID BETWEEN 63601 AND
64800);
SET INTEGRITY FOR STOCK54 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK55.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK55 OFF;
ALTER TABLE STOCK55 DROP CONSTRAINT
STOCK55CKC;
ALTER TABLE STOCK55 ADD CONSTRAINT
STOCK55CKC CHECK (S_W_ID BETWEEN 64801 AND
66000);
SET INTEGRITY FOR STOCK55 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK56.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK56 OFF;
ALTER TABLE STOCK56 DROP CONSTRAINT
STOCK56CKC;
ALTER TABLE STOCK56 ADD CONSTRAINT
STOCK56CKC CHECK (S_W_ID BETWEEN 66001 AND
67200);
SET INTEGRITY FOR STOCK56 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK57.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK57 OFF;
ALTER TABLE STOCK57 DROP CONSTRAINT
STOCK57CKC;
ALTER TABLE STOCK57 ADD CONSTRAINT
STOCK57CKC CHECK (S_W_ID BETWEEN 67201 AND
68400);
SET INTEGRITY FOR STOCK57 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK58.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK58 OFF;
ALTER TABLE STOCK58 DROP CONSTRAINT
STOCK58CKC;
ALTER TABLE STOCK58 ADD CONSTRAINT
STOCK58CKC CHECK (S_W_ID BETWEEN 68401 AND
69600);
SET INTEGRITY FOR STOCK58 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK59.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK59 OFF;
ALTER TABLE STOCK59 DROP CONSTRAINT
STOCK59CKC;
ALTER TABLE STOCK59 ADD CONSTRAINT
STOCK59CKC CHECK (S_W_ID BETWEEN 69601 AND
70800);
SET INTEGRITY FOR STOCK59 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 4801 AND 6000);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK60.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK60 OFF;
ALTER TABLE STOCK60 DROP CONSTRAINT
STOCK60CKC;
ALTER TABLE STOCK60 ADD CONSTRAINT
STOCK60CKC CHECK (S_W_ID BETWEEN 70801 AND
72000);
SET INTEGRITY FOR STOCK60 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK61.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK61 OFF;
ALTER TABLE STOCK61 DROP CONSTRAINT
STOCK61CKC;
ALTER TABLE STOCK61 ADD CONSTRAINT
STOCK61CKC CHECK (S_W_ID BETWEEN 72001 AND
73200);
SET INTEGRITY FOR STOCK61 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK62.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK62 OFF;
ALTER TABLE STOCK62 DROP CONSTRAINT
STOCK62CKC;
ALTER TABLE STOCK62 ADD CONSTRAINT
STOCK62CKC CHECK (S_W_ID BETWEEN 73201 AND
74400);
SET INTEGRITY FOR STOCK62 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK63.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK63 OFF;
ALTER TABLE STOCK63 DROP CONSTRAINT
STOCK63CKC;
ALTER TABLE STOCK63 ADD CONSTRAINT
STOCK63CKC CHECK (S_W_ID BETWEEN 74401 AND
75600);
SET INTEGRITY FOR STOCK63 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK64.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK64 OFF;
ALTER TABLE STOCK64 DROP CONSTRAINT
STOCK64CKC;
ALTER TABLE STOCK64 ADD CONSTRAINT
STOCK64CKC CHECK (S_W_ID BETWEEN 75601 AND
76800);
SET INTEGRITY FOR STOCK64 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK65.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK65 OFF;
ALTER TABLE STOCK65 DROP CONSTRAINT
STOCK65CKC;
ALTER TABLE STOCK65 ADD CONSTRAINT
STOCK65CKC CHECK (S_W_ID BETWEEN 76801 AND
78000);
SET INTEGRITY FOR STOCK65 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK66.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK66 OFF;
ALTER TABLE STOCK66 DROP CONSTRAINT
STOCK66CKC;
ALTER TABLE STOCK66 ADD CONSTRAINT
STOCK66CKC CHECK (S_W_ID BETWEEN 78001 AND
79200);

SET INTEGRITY FOR STOCK66 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK67.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK67 OFF;
ALTER TABLE STOCK67 DROP CONSTRAINT
STOCK67CKC;
ALTER TABLE STOCK67 ADD CONSTRAINT
STOCK67CKC CHECK (S_W_ID BETWEEN 79201 AND
80400);
SET INTEGRITY FOR STOCK67 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK68.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK68 OFF;
ALTER TABLE STOCK68 DROP CONSTRAINT
STOCK68CKC;
ALTER TABLE STOCK68 ADD CONSTRAINT
STOCK68CKC CHECK (S_W_ID BETWEEN 80401 AND
81600);
SET INTEGRITY FOR STOCK68 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST STOCK69.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK69 OFF;
ALTER TABLE STOCK69 DROP CONSTRAINT
STOCK69CKC;
ALTER TABLE STOCK69 ADD CONSTRAINT
STOCK69CKC CHECK (S_W_ID BETWEEN 81601 AND
82800);
SET INTEGRITY FOR STOCK69 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 6001 AND 7200);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK70.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK70 OFF;
ALTER TABLE STOCK70 DROP CONSTRAINT
STOCK70CKC;
ALTER TABLE STOCK70 ADD CONSTRAINT
STOCK70CKC CHECK (S_W_ID BETWEEN 82801 AND
84000);
SET INTEGRITY FOR STOCK70 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK71.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK71 OFF;
ALTER TABLE STOCK71 DROP CONSTRAINT
STOCK71CKC;
ALTER TABLE STOCK71 ADD CONSTRAINT
STOCK71CKC CHECK (S_W_ID BETWEEN 84001 AND
85200);
SET INTEGRITY FOR STOCK71 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK72.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK72 OFF;
ALTER TABLE STOCK72 DROP CONSTRAINT
STOCK72CKC;
ALTER TABLE STOCK72 ADD CONSTRAINT
STOCK72CKC CHECK (S_W_ID BETWEEN 85201 AND
86400);
SET INTEGRITY FOR STOCK72 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK73.ddl

```
connect to TPCC in share mode;
```

```
SET INTEGRITY FOR STOCK73 OFF;
ALTER TABLE STOCK73 DROP CONSTRAINT
STOCK73CKC;
ALTER TABLE STOCK73 ADD CONSTRAINT
STOCK73CKC CHECK (S_W_ID BETWEEN 86401 AND
87600);
SET INTEGRITY FOR STOCK73 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK74.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK74 OFF;
ALTER TABLE STOCK74 DROP CONSTRAINT
STOCK74CKC;
ALTER TABLE STOCK74 ADD CONSTRAINT
STOCK74CKC CHECK (S_W_ID BETWEEN 87601 AND
88800);
SET INTEGRITY FOR STOCK74 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK75.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK75 OFF;
ALTER TABLE STOCK75 DROP CONSTRAINT
STOCK75CKC;
ALTER TABLE STOCK75 ADD CONSTRAINT
STOCK75CKC CHECK (S_W_ID BETWEEN 88801 AND
90000);
SET INTEGRITY FOR STOCK75 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK76.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK76 OFF;
ALTER TABLE STOCK76 DROP CONSTRAINT
STOCK76CKC;
ALTER TABLE STOCK76 ADD CONSTRAINT
STOCK76CKC CHECK (S_W_ID BETWEEN 90001 AND
91200);
SET INTEGRITY FOR STOCK76 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK77.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK77 OFF;
ALTER TABLE STOCK77 DROP CONSTRAINT
STOCK77CKC;
ALTER TABLE STOCK77 ADD CONSTRAINT
STOCK77CKC CHECK (S_W_ID BETWEEN 91201 AND
92400);
SET INTEGRITY FOR STOCK77 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK78.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK78 OFF;
ALTER TABLE STOCK78 DROP CONSTRAINT
STOCK78CKC;
ALTER TABLE STOCK78 ADD CONSTRAINT
STOCK78CKC CHECK (S_W_ID BETWEEN 92401 AND
93600);
SET INTEGRITY FOR STOCK78 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST STOCK79.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK79 OFF;
ALTER TABLE STOCK79 DROP CONSTRAINT
STOCK79CKC;
ALTER TABLE STOCK79 ADD CONSTRAINT
STOCK79CKC CHECK (S_W_ID BETWEEN 93601 AND
94800);
SET INTEGRITY FOR STOCK79 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 7201 AND 8400);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE
UNCHECKED;
connect reset;
```


CRCONST STOCK80.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK80 OFF;
ALTER TABLE STOCK80 DROP CONSTRAINT
STOCK80CKC;
ALTER TABLE STOCK80 ADD CONSTRAINT
STOCK80CKC CHECK (S_W_ID >= 94801);
SET INTEGRITY FOR STOCK80 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 8401 AND 9600);
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 9601 AND 10800);
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 10801
AND 12000);
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 12001
AND 13200);
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 13201
AND 14400);
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 14401
AND 15600);
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 15601
AND 16800);
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 16801
AND 18000);
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 18001
AND 19200);
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 19201
AND 20400);
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 20401
AND 21600);
```

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 21601 AND 22800);
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 1200);
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 22801 AND 24000);
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 24001 AND 25200);
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 25201 AND 26400);
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 26401 AND 27600);
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 27601 AND 28800);
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 28801 AND 30000);
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 30001 AND 31200);
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 31201 AND 32400);
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 32401 AND 33600);
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 33601
AND 34800);
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 1201 AND
2400);
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 34801
AND 36000);
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 36001
AND 37200);

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 BETWEEN 37201
AND 38400);
SET INTEGRITY FOR WAREHOUSE32 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE33.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE33 OFF;
ALTER TABLE WAREHOUSE33 DROP CONSTRAINT
WAREHOUSE33CKC;
ALTER TABLE WAREHOUSE33 ADD CONSTRAINT
WAREHOUSE33CKC CHECK (W_ID BETWEEN 38401
AND 39600);
SET INTEGRITY FOR WAREHOUSE33 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE34.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE34 OFF;
ALTER TABLE WAREHOUSE34 DROP CONSTRAINT
WAREHOUSE34CKC;
ALTER TABLE WAREHOUSE34 ADD CONSTRAINT
WAREHOUSE34CKC CHECK (W_ID BETWEEN 39601
AND 40800);
SET INTEGRITY FOR WAREHOUSE34 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE35.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE35 OFF;

ALTER TABLE WAREHOUSE35 DROP CONSTRAINT
WAREHOUSE35CKC;
ALTER TABLE WAREHOUSE35 ADD CONSTRAINT
WAREHOUSE35CKC CHECK (W_ID BETWEEN 40801
AND 42000);
SET INTEGRITY FOR WAREHOUSE35 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE36.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE36 OFF;
ALTER TABLE WAREHOUSE36 DROP CONSTRAINT
WAREHOUSE36CKC;
ALTER TABLE WAREHOUSE36 ADD CONSTRAINT
WAREHOUSE36CKC CHECK (W_ID BETWEEN 42001
AND 43200);
SET INTEGRITY FOR WAREHOUSE36 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE37.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE37 OFF;
ALTER TABLE WAREHOUSE37 DROP CONSTRAINT
WAREHOUSE37CKC;
ALTER TABLE WAREHOUSE37 ADD CONSTRAINT
WAREHOUSE37CKC CHECK (W_ID BETWEEN 43201
AND 44400);
SET INTEGRITY FOR WAREHOUSE37 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE38.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE38 OFF;
ALTER TABLE WAREHOUSE38 DROP CONSTRAINT
WAREHOUSE38CKC;
ALTER TABLE WAREHOUSE38 ADD CONSTRAINT
WAREHOUSE38CKC CHECK (W_ID BETWEEN 44401
AND 45600);
SET INTEGRITY FOR WAREHOUSE38 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE39.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE39 OFF;
ALTER TABLE WAREHOUSE39 DROP CONSTRAINT
WAREHOUSE39CKC;
ALTER TABLE WAREHOUSE39 ADD CONSTRAINT
WAREHOUSE39CKC CHECK (W_ID BETWEEN 45601
AND 46800);
SET INTEGRITY FOR WAREHOUSE39 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 2401 AND
3600);
SET INTEGRITY FOR WAREHOUSE3 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE40.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE40 OFF;
ALTER TABLE WAREHOUSE40 DROP CONSTRAINT
WAREHOUSE40CKC;
ALTER TABLE WAREHOUSE40 ADD CONSTRAINT
WAREHOUSE40CKC CHECK (W_ID BETWEEN 46801
AND 48000);
SET INTEGRITY FOR WAREHOUSE40 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE41.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE41 OFF;
ALTER TABLE WAREHOUSE41 DROP CONSTRAINT
WAREHOUSE41CKC;
ALTER TABLE WAREHOUSE41 ADD CONSTRAINT
WAREHOUSE41CKC CHECK (W_ID BETWEEN 48001
AND 49200);
SET INTEGRITY FOR WAREHOUSE41 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE42.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE42 OFF;
ALTER TABLE WAREHOUSE42 DROP CONSTRAINT
WAREHOUSE42CKC;
ALTER TABLE WAREHOUSE42 ADD CONSTRAINT
WAREHOUSE42CKC CHECK (W_ID BETWEEN 49201
AND 50400);
SET INTEGRITY FOR WAREHOUSE42 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE43.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE43 OFF;
ALTER TABLE WAREHOUSE43 DROP CONSTRAINT
WAREHOUSE43CKC;
ALTER TABLE WAREHOUSE43 ADD CONSTRAINT
WAREHOUSE43CKC CHECK (W_ID BETWEEN 50401
AND 51600);
SET INTEGRITY FOR WAREHOUSE43 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE44.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE44 OFF;
ALTER TABLE WAREHOUSE44 DROP CONSTRAINT
WAREHOUSE44CKC;
ALTER TABLE WAREHOUSE44 ADD CONSTRAINT
WAREHOUSE44CKC CHECK (W_ID BETWEEN 51601
AND 52800);
SET INTEGRITY FOR WAREHOUSE44 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE45.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE45 OFF;
ALTER TABLE WAREHOUSE45 DROP CONSTRAINT
WAREHOUSE45CKC;
ALTER TABLE WAREHOUSE45 ADD CONSTRAINT
WAREHOUSE45CKC CHECK (W_ID BETWEEN 52801
AND 54000);

SET INTEGRITY FOR WAREHOUSE45 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE46.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE46 OFF;
ALTER TABLE WAREHOUSE46 DROP CONSTRAINT
WAREHOUSE46CKC;
ALTER TABLE WAREHOUSE46 ADD CONSTRAINT
WAREHOUSE46CKC CHECK (W_ID BETWEEN 54001
AND 55200);
SET INTEGRITY FOR WAREHOUSE46 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE47.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE47 OFF;
ALTER TABLE WAREHOUSE47 DROP CONSTRAINT
WAREHOUSE47CKC;
ALTER TABLE WAREHOUSE47 ADD CONSTRAINT
WAREHOUSE47CKC CHECK (W_ID BETWEEN 55201
AND 56400);
SET INTEGRITY FOR WAREHOUSE47 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE48.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE48 OFF;
ALTER TABLE WAREHOUSE48 DROP CONSTRAINT
WAREHOUSE48CKC;
ALTER TABLE WAREHOUSE48 ADD CONSTRAINT
WAREHOUSE48CKC CHECK (W_ID BETWEEN 56401
AND 57600);
SET INTEGRITY FOR WAREHOUSE48 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE49.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE49 OFF;

```
ALTER TABLE WAREHOUSE49 DROP CONSTRAINT
WAREHOUSE49CKC;
ALTER TABLE WAREHOUSE49 ADD CONSTRAINT
WAREHOUSE49CKC CHECK (W_ID BETWEEN 57601
AND 58800);
SET INTEGRITY FOR WAREHOUSE49 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 3601 AND
4800);
SET INTEGRITY FOR WAREHOUSE4 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE50.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE50 OFF;
ALTER TABLE WAREHOUSE50 DROP CONSTRAINT
WAREHOUSE50CKC;
ALTER TABLE WAREHOUSE50 ADD CONSTRAINT
WAREHOUSE50CKC CHECK (W_ID BETWEEN 58801
AND 60000);
SET INTEGRITY FOR WAREHOUSE50 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE51.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE51 OFF;
ALTER TABLE WAREHOUSE51 DROP CONSTRAINT
WAREHOUSE51CKC;
ALTER TABLE WAREHOUSE51 ADD CONSTRAINT
WAREHOUSE51CKC CHECK (W_ID BETWEEN 60001
AND 61200);
SET INTEGRITY FOR WAREHOUSE51 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE52.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE52 OFF;
ALTER TABLE WAREHOUSE52 DROP CONSTRAINT
WAREHOUSE52CKC;
ALTER TABLE WAREHOUSE52 ADD CONSTRAINT
WAREHOUSE52CKC CHECK (W_ID BETWEEN 61201
AND 62400);
SET INTEGRITY FOR WAREHOUSE52 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE53.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE53 OFF;
ALTER TABLE WAREHOUSE53 DROP CONSTRAINT
WAREHOUSE53CKC;
ALTER TABLE WAREHOUSE53 ADD CONSTRAINT
WAREHOUSE53CKC CHECK (W_ID BETWEEN 62401
AND 63600);
SET INTEGRITY FOR WAREHOUSE53 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE54.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE54 OFF;
ALTER TABLE WAREHOUSE54 DROP CONSTRAINT
WAREHOUSE54CKC;
ALTER TABLE WAREHOUSE54 ADD CONSTRAINT
WAREHOUSE54CKC CHECK (W_ID BETWEEN 63601
AND 64800);
SET INTEGRITY FOR WAREHOUSE54 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE55.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE55 OFF;
ALTER TABLE WAREHOUSE55 DROP CONSTRAINT
WAREHOUSE55CKC;
ALTER TABLE WAREHOUSE55 ADD CONSTRAINT
WAREHOUSE55CKC CHECK (W_ID BETWEEN 64801
AND 66000);
SET INTEGRITY FOR WAREHOUSE55 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE56.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE56 OFF;
ALTER TABLE WAREHOUSE56 DROP CONSTRAINT
WAREHOUSE56CKC;
ALTER TABLE WAREHOUSE56 ADD CONSTRAINT
WAREHOUSE56CKC CHECK (W_ID BETWEEN 66001
AND 67200);
SET INTEGRITY FOR WAREHOUSE56 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE57.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE57 OFF;
ALTER TABLE WAREHOUSE57 DROP CONSTRAINT
WAREHOUSE57CKC;
ALTER TABLE WAREHOUSE57 ADD CONSTRAINT
WAREHOUSE57CKC CHECK (W_ID BETWEEN 67201
AND 68400);
SET INTEGRITY FOR WAREHOUSE57 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE58.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE58 OFF;
ALTER TABLE WAREHOUSE58 DROP CONSTRAINT
WAREHOUSE58CKC;
ALTER TABLE WAREHOUSE58 ADD CONSTRAINT
WAREHOUSE58CKC CHECK (W_ID BETWEEN 68401
AND 69600);
SET INTEGRITY FOR WAREHOUSE58 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE59.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE59 OFF;
ALTER TABLE WAREHOUSE59 DROP CONSTRAINT
WAREHOUSE59CKC;
ALTER TABLE WAREHOUSE59 ADD CONSTRAINT
WAREHOUSE59CKC CHECK (W_ID BETWEEN 69601
AND 70800);
```

SET INTEGRITY FOR WAREHOUSE59 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 4801 AND 6000);
SET INTEGRITY FOR WAREHOUSE5 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE60.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE60 OFF;
ALTER TABLE WAREHOUSE60 DROP CONSTRAINT WAREHOUSE60CKC;
ALTER TABLE WAREHOUSE60 ADD CONSTRAINT WAREHOUSE60CKC CHECK (W_ID BETWEEN 70801 AND 72000);
SET INTEGRITY FOR WAREHOUSE60 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE61.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE61 OFF;
ALTER TABLE WAREHOUSE61 DROP CONSTRAINT WAREHOUSE61CKC;
ALTER TABLE WAREHOUSE61 ADD CONSTRAINT WAREHOUSE61CKC CHECK (W_ID BETWEEN 72001 AND 73200);
SET INTEGRITY FOR WAREHOUSE61 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE62.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE62 OFF;

ALTER TABLE WAREHOUSE62 DROP CONSTRAINT WAREHOUSE62CKC;
ALTER TABLE WAREHOUSE62 ADD CONSTRAINT WAREHOUSE62CKC CHECK (W_ID BETWEEN 73201 AND 74400);
SET INTEGRITY FOR WAREHOUSE62 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE63.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE63 OFF;
ALTER TABLE WAREHOUSE63 DROP CONSTRAINT WAREHOUSE63CKC;
ALTER TABLE WAREHOUSE63 ADD CONSTRAINT WAREHOUSE63CKC CHECK (W_ID BETWEEN 74401 AND 75600);
SET INTEGRITY FOR WAREHOUSE63 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE64.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE64 OFF;
ALTER TABLE WAREHOUSE64 DROP CONSTRAINT WAREHOUSE64CKC;
ALTER TABLE WAREHOUSE64 ADD CONSTRAINT WAREHOUSE64CKC CHECK (W_ID BETWEEN 75601 AND 76800);
SET INTEGRITY FOR WAREHOUSE64 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE65.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE65 OFF;
ALTER TABLE WAREHOUSE65 DROP CONSTRAINT WAREHOUSE65CKC;
ALTER TABLE WAREHOUSE65 ADD CONSTRAINT WAREHOUSE65CKC CHECK (W_ID BETWEEN 76801 AND 78000);
SET INTEGRITY FOR WAREHOUSE65 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE66.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE66 OFF;
ALTER TABLE WAREHOUSE66 DROP CONSTRAINT WAREHOUSE66CKC;
ALTER TABLE WAREHOUSE66 ADD CONSTRAINT WAREHOUSE66CKC CHECK (W_ID BETWEEN 78001 AND 79200);
SET INTEGRITY FOR WAREHOUSE66 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE67.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE67 OFF;
ALTER TABLE WAREHOUSE67 DROP CONSTRAINT WAREHOUSE67CKC;
ALTER TABLE WAREHOUSE67 ADD CONSTRAINT WAREHOUSE67CKC CHECK (W_ID BETWEEN 79201 AND 80400);
SET INTEGRITY FOR WAREHOUSE67 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE68.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE68 OFF;
ALTER TABLE WAREHOUSE68 DROP CONSTRAINT WAREHOUSE68CKC;
ALTER TABLE WAREHOUSE68 ADD CONSTRAINT WAREHOUSE68CKC CHECK (W_ID BETWEEN 80401 AND 81600);
SET INTEGRITY FOR WAREHOUSE68 ALL IMMEDIATE UNCHECKED;
connect reset;

CRCONST WAREHOUSE69.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE69 OFF;
ALTER TABLE WAREHOUSE69 DROP CONSTRAINT WAREHOUSE69CKC;
ALTER TABLE WAREHOUSE69 ADD CONSTRAINT WAREHOUSE69CKC CHECK (W_ID BETWEEN 81601 AND 82800);
SET INTEGRITY FOR WAREHOUSE69 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 6001 AND
7200);
SET INTEGRITY FOR WAREHOUSE6 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE70.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE70 OFF;
ALTER TABLE WAREHOUSE70 DROP CONSTRAINT
WAREHOUSE70CKC;
ALTER TABLE WAREHOUSE70 ADD CONSTRAINT
WAREHOUSE70CKC CHECK (W_ID BETWEEN 82801
AND 84000);
SET INTEGRITY FOR WAREHOUSE70 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE71.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE71 OFF;
ALTER TABLE WAREHOUSE71 DROP CONSTRAINT
WAREHOUSE71CKC;
ALTER TABLE WAREHOUSE71 ADD CONSTRAINT
WAREHOUSE71CKC CHECK (W_ID BETWEEN 84001
AND 85200);
SET INTEGRITY FOR WAREHOUSE71 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE72.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE72 OFF;
ALTER TABLE WAREHOUSE72 DROP CONSTRAINT
WAREHOUSE72CKC;
ALTER TABLE WAREHOUSE72 ADD CONSTRAINT
WAREHOUSE72CKC CHECK (W_ID BETWEEN 85201
AND 86400);

SET INTEGRITY FOR WAREHOUSE72 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE73.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE73 OFF;
ALTER TABLE WAREHOUSE73 DROP CONSTRAINT
WAREHOUSE73CKC;
ALTER TABLE WAREHOUSE73 ADD CONSTRAINT
WAREHOUSE73CKC CHECK (W_ID BETWEEN 86401
AND 87600);
SET INTEGRITY FOR WAREHOUSE73 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE74.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE74 OFF;
ALTER TABLE WAREHOUSE74 DROP CONSTRAINT
WAREHOUSE74CKC;
ALTER TABLE WAREHOUSE74 ADD CONSTRAINT
WAREHOUSE74CKC CHECK (W_ID BETWEEN 87601
AND 88800);
SET INTEGRITY FOR WAREHOUSE74 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE75.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE75 OFF;
ALTER TABLE WAREHOUSE75 DROP CONSTRAINT
WAREHOUSE75CKC;
ALTER TABLE WAREHOUSE75 ADD CONSTRAINT
WAREHOUSE75CKC CHECK (W_ID BETWEEN 88801
AND 90000);
SET INTEGRITY FOR WAREHOUSE75 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE76.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE76 OFF;

ALTER TABLE WAREHOUSE76 DROP CONSTRAINT
WAREHOUSE76CKC;
ALTER TABLE WAREHOUSE76 ADD CONSTRAINT
WAREHOUSE76CKC CHECK (W_ID BETWEEN 90001
AND 91200);
SET INTEGRITY FOR WAREHOUSE76 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE77.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE77 OFF;
ALTER TABLE WAREHOUSE77 DROP CONSTRAINT
WAREHOUSE77CKC;
ALTER TABLE WAREHOUSE77 ADD CONSTRAINT
WAREHOUSE77CKC CHECK (W_ID BETWEEN 91201
AND 92400);
SET INTEGRITY FOR WAREHOUSE77 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE78.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE78 OFF;
ALTER TABLE WAREHOUSE78 DROP CONSTRAINT
WAREHOUSE78CKC;
ALTER TABLE WAREHOUSE78 ADD CONSTRAINT
WAREHOUSE78CKC CHECK (W_ID BETWEEN 92401
AND 93600);
SET INTEGRITY FOR WAREHOUSE78 ALL IMMEDIATE
UNCHECKED;
connect reset;

CRCONST WAREHOUSE79.ddl

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE79 OFF;
ALTER TABLE WAREHOUSE79 DROP CONSTRAINT
WAREHOUSE79CKC;
ALTER TABLE WAREHOUSE79 ADD CONSTRAINT
WAREHOUSE79CKC CHECK (W_ID BETWEEN 93601
AND 94800);
SET INTEGRITY FOR WAREHOUSE79 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 7201 AND
8400);
SET INTEGRITY FOR WAREHOUSE7 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

CRCONST WAREHOUSE80.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE80 OFF;
ALTER TABLE WAREHOUSE80 DROP CONSTRAINT
WAREHOUSE80CKC;
ALTER TABLE WAREHOUSE80 ADD CONSTRAINT
WAREHOUSE80CKC CHECK (W_ID >= 94801);
SET INTEGRITY FOR WAREHOUSE80 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 8401 AND
9600);
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 9601 AND
10800);
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_IDXB33.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB33;
CREATE INDEX CUST_IDXB33
      ON CUSTOMER33(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB34.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB34;
CREATE INDEX CUST_IDXB34
    ON CUSTOMER34(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB35.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB35;
CREATE INDEX CUST_IDXB35
    ON CUSTOMER35(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB36.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB36;
CREATE INDEX CUST_IDXB36
    ON CUSTOMER36(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB37.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB37;
CREATE INDEX CUST_IDXB37
    ON CUSTOMER37(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB38.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB38;
CREATE INDEX CUST_IDXB38
    ON CUSTOMER38(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB39.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB39;
CREATE INDEX CUST_IDXB39
    ON CUSTOMER39(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

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_IDXB40.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB40;
CREATE INDEX CUST_IDXB40
    ON CUSTOMER40(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB41.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB41;
CREATE INDEX CUST_IDXB41
    ON CUSTOMER41(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB42.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB42;
CREATE INDEX CUST_IDXB42
    ON CUSTOMER42(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB43.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB43;
```

```
CREATE INDEX CUST_IDXB43
    ON CUSTOMER43(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB44.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB44;
CREATE INDEX CUST_IDXB44
    ON CUSTOMER44(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB45.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB45;
CREATE INDEX CUST_IDXB45
    ON CUSTOMER45(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB46.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB46;
CREATE INDEX CUST_IDXB46
    ON CUSTOMER46(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB47.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB47;
CREATE INDEX CUST_IDXB47
    ON CUSTOMER47(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB48.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB48;
CREATE INDEX CUST_IDXB48
```

ON CUSTOMER48(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB49.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB49;
CREATE INDEX CUST_IDXB49
ON CUSTOMER49(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

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_IDXB50.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB50;
CREATE INDEX CUST_IDXB50
ON CUSTOMER50(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB51.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB51;
CREATE INDEX CUST_IDXB51
ON CUSTOMER51(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB52.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB52;
CREATE INDEX CUST_IDXB52
ON CUSTOMER52(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;

connect reset;

CRIDX_CUST_IDXB53.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB53;
CREATE INDEX CUST_IDXB53
ON CUSTOMER53(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB54.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB54;
CREATE INDEX CUST_IDXB54
ON CUSTOMER54(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB55.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB55;
CREATE INDEX CUST_IDXB55
ON CUSTOMER55(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB56.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB56;
CREATE INDEX CUST_IDXB56
ON CUSTOMER56(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB57.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB57;
CREATE INDEX CUST_IDXB57
ON CUSTOMER57(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB58.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB58;
CREATE INDEX CUST_IDXB58
ON CUSTOMER58(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB59.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB59;
CREATE INDEX CUST_IDXB59
ON CUSTOMER59(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

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_IDXB60.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB60;
CREATE INDEX CUST_IDXB60
ON CUSTOMER60(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB61.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB61;
CREATE INDEX CUST_IDXB61
ON CUSTOMER61(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB62.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB62;
CREATE INDEX CUST_IDXB62
      ON CUSTOMER62(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB63.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB63;
CREATE INDEX CUST_IDXB63
      ON CUSTOMER63(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB64.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB64;
CREATE INDEX CUST_IDXB64
      ON CUSTOMER64(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB65.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB65;
CREATE INDEX CUST_IDXB65
      ON CUSTOMER65(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB66.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB66;
CREATE INDEX CUST_IDXB66
      ON CUSTOMER66(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB67.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB67;
```

```
CREATE INDEX CUST_IDXB67
      ON CUSTOMER67(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB68.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB68;
CREATE INDEX CUST_IDXB68
      ON CUSTOMER68(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB69.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB69;
CREATE INDEX CUST_IDXB69
      ON CUSTOMER69(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

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_IDXB70.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB70;
CREATE INDEX CUST_IDXB70
      ON CUSTOMER70(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB71.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB71;
CREATE INDEX CUST_IDXB71
```

```
      ON CUSTOMER71(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB72.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB72;
CREATE INDEX CUST_IDXB72
      ON CUSTOMER72(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB73.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB73;
CREATE INDEX CUST_IDXB73
      ON CUSTOMER73(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB74.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB74;
CREATE INDEX CUST_IDXB74
      ON CUSTOMER74(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB75.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB75;
CREATE INDEX CUST_IDXB75
      ON CUSTOMER75(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

CRIDX_CUST_IDXB76.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB76;
CREATE INDEX CUST_IDXB76
      ON CUSTOMER76(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
```

connect reset;

CRIDX_CUST_IDXB77.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB77;
CREATE INDEX CUST_IDXB77
ON CUSTOMER77(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB78.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB78;
CREATE INDEX CUST_IDXB78
ON CUSTOMER78(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

CRIDX_CUST_IDXB79.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB79;
CREATE INDEX CUST_IDXB79
ON CUSTOMER79(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

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_IDXB80.ddl

connect to TPCC in share mode;
DROP INDEX CUST_IDXB80;
CREATE INDEX CUST_IDXB80
ON CUSTOMER80(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

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_ORDER_IDXB10.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB10;
CREATE INDEX ORDER_IDXB10
ON ORDERS10(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB11.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB11;
CREATE INDEX ORDER_IDXB11
ON ORDERS11(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB12.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB12;
CREATE INDEX ORDER_IDXB12
ON ORDERS12(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB13.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB13;
CREATE INDEX ORDER_IDXB13
ON ORDERS13(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB14.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB14;
CREATE INDEX ORDER_IDXB14
ON ORDERS14(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB15.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB15;
CREATE INDEX ORDER_IDXB15
ON ORDERS15(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB16.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB16;
CREATE INDEX ORDER_IDXB16
ON ORDERS16(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB17.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB17;
CREATE INDEX ORDER_IDXB17
ON ORDERS17(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDER_IDXB18.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_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_IDXB33.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB33;
CREATE INDEX ORDR_IDXB33
ON ORDERS33(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB34.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB34;
CREATE INDEX ORDR_IDXB34
ON ORDERS34(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB35.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB35;
CREATE INDEX ORDR_IDXB35
ON ORDERS35(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB36.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB36;
CREATE INDEX ORDR_IDXB36
ON ORDERS36(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB37.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB37;
CREATE INDEX ORDR_IDXB37
ON ORDERS37(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB38.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB38;
CREATE INDEX ORDR_IDXB38
ON ORDERS38(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB39.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB39;
CREATE INDEX ORDR_IDXB39
ON ORDERS39(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

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_IDXB40.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB40;
CREATE INDEX ORDR_IDXB40
ON ORDERS40(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB41.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB41;

CREATE INDEX ORDR_IDXB41
ON ORDERS41(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB42.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB42;
CREATE INDEX ORDR_IDXB42
ON ORDERS42(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB43.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB43;
CREATE INDEX ORDR_IDXB43
ON ORDERS43(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB44.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB44;
CREATE INDEX ORDR_IDXB44
ON ORDERS44(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB45.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB45;
CREATE INDEX ORDR_IDXB45
ON ORDERS45(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX_ORDR_IDXB46.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB46;
CREATE INDEX ORDR_IDXB46

ON ORDERS46(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX47.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX47;
CREATE INDEX ORDER_IDX47
ON ORDERS47(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX48.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX48;
CREATE INDEX ORDER_IDX48
ON ORDERS48(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX49.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX49;
CREATE INDEX ORDER_IDX49
ON ORDERS49(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX4.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX4;
CREATE INDEX ORDER_IDX4
ON ORDERS4(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX50.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX50;
CREATE INDEX ORDER_IDX50
ON ORDERS50(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;

connect reset;

CRIDX ORDER IDX51.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX51;
CREATE INDEX ORDER_IDX51
ON ORDERS51(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX52.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX52;
CREATE INDEX ORDER_IDX52
ON ORDERS52(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX53.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX53;
CREATE INDEX ORDER_IDX53
ON ORDERS53(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX54.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX54;
CREATE INDEX ORDER_IDX54
ON ORDERS54(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX55.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX55;
CREATE INDEX ORDER_IDX55
ON ORDERS55(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX56.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX56;
CREATE INDEX ORDER_IDX56
ON ORDERS56(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX57.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX57;
CREATE INDEX ORDER_IDX57
ON ORDERS57(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX58.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX58;
CREATE INDEX ORDER_IDX58
ON ORDERS58(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX59.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX59;
CREATE INDEX ORDER_IDX59
ON ORDERS59(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX5.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDX5;
CREATE INDEX ORDER_IDX5
ON ORDERS5(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX60.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB60;
CREATE INDEX ORDR_IDXB60
ON ORDERS60(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB61.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB61;
CREATE INDEX ORDR_IDXB61
ON ORDERS61(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB62.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB62;
CREATE INDEX ORDR_IDXB62
ON ORDERS62(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB63.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB63;
CREATE INDEX ORDR_IDXB63
ON ORDERS63(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB64.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB64;
CREATE INDEX ORDR_IDXB64
ON ORDERS64(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB65.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB65;

CREATE INDEX ORDR_IDXB65
ON ORDERS65(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB66.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB66;
CREATE INDEX ORDR_IDXB66
ON ORDERS66(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB67.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB67;
CREATE INDEX ORDR_IDXB67
ON ORDERS67(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB68.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB68;
CREATE INDEX ORDR_IDXB68
ON ORDERS68(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB69.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB69;
CREATE INDEX ORDR_IDXB69
ON ORDERS69(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

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_IDXB70.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB70;
CREATE INDEX ORDR_IDXB70
ON ORDERS70(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB71.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB71;
CREATE INDEX ORDR_IDXB71
ON ORDERS71(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB72.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB72;
CREATE INDEX ORDR_IDXB72
ON ORDERS72(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB73.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB73;
CREATE INDEX ORDR_IDXB73
ON ORDERS73(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDR_IDXB74.ddl

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB74;
CREATE INDEX ORDR_IDXB74
ON ORDERS74(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;

connect reset;

CRIDX ORDER IDX75.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB75;
CREATE INDEX ORDER_IDXB75
ON ORDERS75(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX76.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB76;
CREATE INDEX ORDER_IDXB76
ON ORDERS76(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX77.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB77;
CREATE INDEX ORDER_IDXB77
ON ORDERS77(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX78.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB78;
CREATE INDEX ORDER_IDXB78
ON ORDERS78(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX79.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB79;
CREATE INDEX ORDER_IDXB79
ON ORDERS79(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX77.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB77;
CREATE INDEX ORDER_IDXB77
ON ORDERS77(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX80.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB80;
CREATE INDEX ORDER_IDXB80
ON ORDERS80(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX88.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB88;
CREATE INDEX ORDER_IDXB88
ON ORDERS88(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

CRIDX ORDER IDX89.ddl

connect to TPCC in share mode;
DROP INDEX ORDER_IDXB89;
CREATE INDEX ORDER_IDXB89
ON ORDERS89(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 10801 ENDING AT
12000,
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 12001 ENDING AT
13200,
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 13201 ENDING AT
14400,
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 14401 ENDING AT
15600,
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 15601 ENDING AT
16800,
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 16801 ENDING AT
18000,
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 18001 ENDING AT
19200,
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 19201 ENDING AT
20400,
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 20401 ENDING AT
21600,
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 21601 ENDING AT

22800,

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 1200,

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 22801 ENDING AT

```

24000,

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 24001 ENDING AT

25200,

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 25201 ENDING AT
26400,
  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 26401 ENDING AT
27600,
  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 27601 ENDING AT
28800,
  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 28801 ENDING AT
30000,
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 30001 ENDING AT
31200,
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 31201 ENDING AT
32400,
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 32401 ENDING AT
33600,
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,

```

NULL, C_YTD_PAYMENT DECIMAL(12,2) NOT
)
 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 33601 ENDING AT
 34800,
 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 1201 ENDING AT
 2400,
 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 34801 ENDING AT

36000,
 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 36001 ENDING AT

37200,
 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 37201 ENDING AT
38400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER33.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER33;
CREATE TABLE CUSTOMER33
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

)
IN ts_customer_033
INDEX IN is_customer_033
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 38401 ENDING AT
39600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER34.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER34;
CREATE TABLE CUSTOMER34
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_034
INDEX IN is_customer_034
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 39601 ENDING AT
40800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER35.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER35;
CREATE TABLE CUSTOMER35
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_035
INDEX IN is_customer_035
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 40801 ENDING AT
42000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER36.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER36;
CREATE TABLE CUSTOMER36
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,

```

```

C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_036
INDEX IN is_customer_036
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 42001 ENDING AT
43200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER37.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER37;
CREATE TABLE CUSTOMER37

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_037
INDEX IN is_customer_037
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 43201 ENDING AT
44400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER38.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER38;
CREATE TABLE CUSTOMER38

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_038
INDEX IN is_customer_038
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 44401 ENDING AT
45600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER39.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER39;
CREATE TABLE CUSTOMER39

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,

```

```

C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_039
INDEX IN is_customer_039
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 45601 ENDING AT
46800,
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 2401 ENDING AT
3600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER40.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER40;
CREATE TABLE CUSTOMER40
(
  C_ID          INTEGER    NOT NULL,
  C_STATE      CHAR(2)    NOT NULL,
  C_ZIP        CHAR(9)    NOT NULL,
  C_PHONE      CHAR(16)   NOT NULL,
  C_SINCE      TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)    NOT NULL,
  C_CREDIT     CHAR(2)    NOT NULL,

```

```

  C_DISCOUNT  REAL       NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16) NOT NULL,
  C_FIRST      VARCHAR(16) NOT NULL,
  C_STREET_1   VARCHAR(20) NOT NULL,
  C_STREET_2   VARCHAR(20) NOT NULL,
  C_CITY       VARCHAR(20) NOT NULL,
  C_D_ID       SMALLINT   NOT NULL,
  C_W_ID       INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_040
INDEX IN is_customer_040
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 46801 ENDING AT
48000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER41.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER41;
CREATE TABLE CUSTOMER41
(
  C_ID          INTEGER    NOT NULL,
  C_STATE      CHAR(2)    NOT NULL,
  C_ZIP        CHAR(9)    NOT NULL,
  C_PHONE      CHAR(16)   NOT NULL,
  C_SINCE      TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)    NOT NULL,
  C_CREDIT     CHAR(2)    NOT NULL,
  C_DISCOUNT  REAL       NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16) NOT NULL,
  C_FIRST      VARCHAR(16) NOT NULL,
  C_STREET_1   VARCHAR(20) NOT NULL,
  C_STREET_2   VARCHAR(20) NOT NULL,
  C_CITY       VARCHAR(20) NOT NULL,
  C_D_ID       SMALLINT   NOT NULL,
  C_W_ID       INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,

```

```

  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_041
INDEX IN is_customer_041
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 48001 ENDING AT
49200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER42.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER42;
CREATE TABLE CUSTOMER42
(
  C_ID          INTEGER    NOT NULL,
  C_STATE      CHAR(2)    NOT NULL,
  C_ZIP        CHAR(9)    NOT NULL,
  C_PHONE      CHAR(16)   NOT NULL,
  C_SINCE      TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)    NOT NULL,
  C_CREDIT     CHAR(2)    NOT NULL,
  C_DISCOUNT  REAL       NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16) NOT NULL,
  C_FIRST      VARCHAR(16) NOT NULL,
  C_STREET_1   VARCHAR(20) NOT NULL,
  C_STREET_2   VARCHAR(20) NOT NULL,
  C_CITY       VARCHAR(20) NOT NULL,
  C_D_ID       SMALLINT   NOT NULL,
  C_W_ID       INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_042
INDEX IN is_customer_042
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 49201 ENDING AT
50400,
  C_D_ID STARTING FROM 1 ENDING AT 10

```

```
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER43.ddl

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER43;
CREATE TABLE CUSTOMER43
```

```
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_043
INDEX IN is_customer_043
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 50401 ENDING AT
51600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER44.ddl

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER44;
CREATE TABLE CUSTOMER44
```

```
(
```

```
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_044
INDEX IN is_customer_044
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 51601 ENDING AT
52800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER45.ddl

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER45;
CREATE TABLE CUSTOMER45
```

```
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
```

```
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_045
INDEX IN is_customer_045
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 52801 ENDING AT
54000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER46.ddl

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER46;
CREATE TABLE CUSTOMER46
```

```
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
NULL,
```

```

)
IN ts_customer_046
INDEX IN is_customer_046
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 54001 ENDING AT
55200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB CUSTOMER47.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER47;
CREATE TABLE CUSTOMER47
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)

```

```

IN ts_customer_047
INDEX IN is_customer_047
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 55201 ENDING AT
56400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB CUSTOMER48.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER48;
CREATE TABLE CUSTOMER48
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_048
INDEX IN is_customer_048
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 56401 ENDING AT
57600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB CUSTOMER49.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER49;
CREATE TABLE CUSTOMER49
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,

```

```

  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_049
INDEX IN is_customer_049
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 57601 ENDING AT
58800,
  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 3601 ENDING AT
4800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER50.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER50;
CREATE TABLE CUSTOMER50
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_050
INDEX IN is_customer_050
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 58801 ENDING AT
60000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER51.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER51;
CREATE TABLE CUSTOMER51
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_051
INDEX IN is_customer_051
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 60001 ENDING AT
61200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER52.ddl

```

connect to TPCC in share mode;

```

```

DROP TABLE CUSTOMER52;
CREATE TABLE CUSTOMER52
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_052
INDEX IN is_customer_052
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 61201 ENDING AT
62400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER53.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER53;
CREATE TABLE CUSTOMER53
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,

```

```

C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_053
INDEX IN is_customer_053
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 62401 ENDING AT
63600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER54.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER54;
CREATE TABLE CUSTOMER54

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,

```

```

NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_054
INDEX IN is_customer_054
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 63601 ENDING AT
64800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER55.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER55;
CREATE TABLE CUSTOMER55

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_055
INDEX IN is_customer_055
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 64801 ENDING AT
66000,
C_D_ID STARTING FROM 1 ENDING AT 10

```

```

NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_056
INDEX IN is_customer_056
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 66001 ENDING AT
67200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER56.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER56;
CREATE TABLE CUSTOMER56

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_056
INDEX IN is_customer_056
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 66001 ENDING AT
67200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_056
INDEX IN is_customer_056
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 66001 ENDING AT
67200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER57.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER57;
CREATE TABLE CUSTOMER57

```

```

(

```

```

C_ID      INTEGER      NOT NULL,
C_STATE   CHAR(2)      NOT NULL,
C_ZIP     CHAR(9)      NOT NULL,
C_PHONE   CHAR(16)     NOT NULL,
C_SINCE   TIMESTAMP   NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE  CHAR(2)      NOT NULL,
C_CREDIT  CHAR(2)      NOT NULL,
C_DISCOUNT REAL      NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16)  NOT NULL,
C_FIRST   VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20)  NOT NULL,
C_D_ID    SMALLINT    NOT NULL,
C_W_ID    INTEGER      NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT

```

```

NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)

```

```

IN ts_customer_057
INDEX IN is_customer_057
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 67201 ENDING AT

```

```

68400,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER58.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER58;
CREATE TABLE CUSTOMER58

```

```

(
C_ID      INTEGER      NOT NULL,
C_STATE   CHAR(2)      NOT NULL,
C_ZIP     CHAR(9)      NOT NULL,
C_PHONE   CHAR(16)     NOT NULL,
C_SINCE   TIMESTAMP   NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE  CHAR(2)      NOT NULL,
C_CREDIT  CHAR(2)      NOT NULL,
C_DISCOUNT REAL      NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16)  NOT NULL,

```

```

C_FIRST   VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20)  NOT NULL,
C_D_ID    SMALLINT    NOT NULL,
C_W_ID    INTEGER      NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT

```

```

NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)

```

```

IN ts_customer_058
INDEX IN is_customer_058
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 68401 ENDING AT

```

```

69600,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER59.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER59;
CREATE TABLE CUSTOMER59

```

```

(
C_ID      INTEGER      NOT NULL,
C_STATE   CHAR(2)      NOT NULL,
C_ZIP     CHAR(9)      NOT NULL,
C_PHONE   CHAR(16)     NOT NULL,
C_SINCE   TIMESTAMP   NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE  CHAR(2)      NOT NULL,
C_CREDIT  CHAR(2)      NOT NULL,
C_DISCOUNT REAL      NOT NULL,
C_DATA    VARCHAR(500) NOT NULL,
C_LAST    VARCHAR(16)  NOT NULL,
C_FIRST   VARCHAR(16)  NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20)  NOT NULL,
C_D_ID    SMALLINT    NOT NULL,
C_W_ID    INTEGER      NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT

```

```

NULL,
C_PAYMENT_CNT INTEGER  NOT NULL
)

```

```

)
IN ts_customer_059
INDEX IN is_customer_059
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 69601 ENDING AT

```

```

70800,
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 4801 ENDING AT

```

```

6000,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;

```


CRTB_CUSTOMER60.ddl

connect to TPCC in share mode;
DROP TABLE CUSTOMER60;
CREATE TABLE CUSTOMER60

```
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_060
INDEX IN is_customer_060
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 70801 ENDING AT
72000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER61.ddl

connect to TPCC in share mode;
DROP TABLE CUSTOMER61;
CREATE TABLE CUSTOMER61

```
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
```

```
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_061
INDEX IN is_customer_061
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 72001 ENDING AT
73200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER62.ddl

connect to TPCC in share mode;
DROP TABLE CUSTOMER62;
CREATE TABLE CUSTOMER62

```
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
```

```
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_062
INDEX IN is_customer_062
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 73201 ENDING AT
74400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_CUSTOMER63.ddl

connect to TPCC in share mode;
DROP TABLE CUSTOMER63;
CREATE TABLE CUSTOMER63

```
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT REAL      NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_063
INDEX IN is_customer_063
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
```

```

C_W_ID STARTING FROM 74401 ENDING AT
75600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER64.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER64;
CREATE TABLE CUSTOMER64
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_064
INDEX IN is_customer_064
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 75601 ENDING AT
76800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER65.ddl

```

connect to TPCC in share mode;

```

```

DROP TABLE CUSTOMER65;
CREATE TABLE CUSTOMER65
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT
NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_065
INDEX IN is_customer_065
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 76801 ENDING AT
78000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER66.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER66;
CREATE TABLE CUSTOMER66
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,

```

```

C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_066
INDEX IN is_customer_066
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 78001 ENDING AT
79200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER67.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER67;
CREATE TABLE CUSTOMER67
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,

```

```

NULL, C_YTD_PAYMENT DECIMAL(12,2) NOT
      C_PAYMENT_CNT INTEGER NOT NULL
      )
      IN ts_customer_067
      INDEX IN is_customer_067
      ORGANIZE BY KEY SEQUENCE (
      C_ID STARTING FROM 1 ENDING AT 3000,
      C_W_ID STARTING FROM 79201 ENDING AT
80400, C_D_ID STARTING FROM 1 ENDING AT 10
      )
      ALLOW OVERFLOW;
connect reset;

```

CRTB CUSTOMER68.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER68;
CREATE TABLE CUSTOMER68

```

```

(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT

```

```

NULL, C_PAYMENT_CNT INTEGER NOT NULL
      )
      IN ts_customer_068
      INDEX IN is_customer_068
      ORGANIZE BY KEY SEQUENCE (
      C_ID STARTING FROM 1 ENDING AT 3000,
      C_W_ID STARTING FROM 80401 ENDING AT
81600, C_D_ID STARTING FROM 1 ENDING AT 10

```

```

      )
      ALLOW OVERFLOW;
connect reset;

```

CRTB CUSTOMER69.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER69;
CREATE TABLE CUSTOMER69

```

```

(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT

```

```

NULL, C_PAYMENT_CNT INTEGER NOT NULL
      )
      IN ts_customer_069
      INDEX IN is_customer_069
      ORGANIZE BY KEY SEQUENCE (
      C_ID STARTING FROM 1 ENDING AT 3000,
      C_W_ID STARTING FROM 81601 ENDING AT
82800, C_D_ID STARTING FROM 1 ENDING AT 10
      )
      ALLOW OVERFLOW;
connect reset;

```

```

82800, 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 6001 ENDING AT
7200, C_D_ID STARTING FROM 1 ENDING AT 10
      )
      ALLOW OVERFLOW;
connect reset;

```

```

7200, C_D_ID STARTING FROM 1 ENDING AT 10
      )
      ALLOW OVERFLOW;
connect reset;

```

CRTB CUSTOMER70.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER70;
CREATE TABLE CUSTOMER70

```

```

(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,

```

```

C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_070
INDEX IN is_customer_070
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 82801 ENDING AT
84000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER71.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER71;
CREATE TABLE CUSTOMER71
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

)
IN ts_customer_071
INDEX IN is_customer_071
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 84001 ENDING AT
85200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER72.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER72;
CREATE TABLE CUSTOMER72
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

IN ts_customer_072
INDEX IN is_customer_072
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 85201 ENDING AT
86400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER73.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER73;
CREATE TABLE CUSTOMER73
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_073
INDEX IN is_customer_073
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 86401 ENDING AT
87600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER74.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER74;
CREATE TABLE CUSTOMER74
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,

```

```

C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_074
INDEX IN is_customer_074
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 87601 ENDING AT
88800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER75.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER75;
CREATE TABLE CUSTOMER75

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_075
INDEX IN is_customer_075
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 88801 ENDING AT
90000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER76.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER76;
CREATE TABLE CUSTOMER76

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_076
INDEX IN is_customer_076
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 90001 ENDING AT
91200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER77.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER77;
CREATE TABLE CUSTOMER77

```

```

(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,

```

```

C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_077
INDEX IN is_customer_077
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 91201 ENDING AT
92400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER78.ddl

```

connect to TPCC in share mode;

```

```

DROP TABLE CUSTOMER78;
CREATE TABLE CUSTOMER78
(
  C_ID          INTEGER    NOT NULL,
  C_STATE      CHAR(2)    NOT NULL,
  C_ZIP        CHAR(9)    NOT NULL,
  C_PHONE      CHAR(16)   NOT NULL,
  C_SINCE      TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)    NOT NULL,
  C_CREDIT     CHAR(2)    NOT NULL,
  C_DISCOUNT  REAL       NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16) NOT NULL,
  C_FIRST      VARCHAR(16) NOT NULL,
  C_STREET_1   VARCHAR(20) NOT NULL,
  C_STREET_2   VARCHAR(20) NOT NULL,
  C_CITY       VARCHAR(20) NOT NULL,
  C_D_ID       SMALLINT   NOT NULL,
  C_W_ID       INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_078
INDEX IN is_customer_078
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 92401 ENDING AT
93600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER79.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER79;
CREATE TABLE CUSTOMER79
(
  C_ID          INTEGER    NOT NULL,
  C_STATE      CHAR(2)    NOT NULL,
  C_ZIP        CHAR(9)    NOT NULL,
  C_PHONE      CHAR(16)   NOT NULL,
  C_SINCE      TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)    NOT NULL,
  C_CREDIT     CHAR(2)    NOT NULL,

```

```

  C_DISCOUNT  REAL       NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16) NOT NULL,
  C_FIRST      VARCHAR(16) NOT NULL,
  C_STREET_1   VARCHAR(20) NOT NULL,
  C_STREET_2   VARCHAR(20) NOT NULL,
  C_CITY       VARCHAR(20) NOT NULL,
  C_D_ID       SMALLINT   NOT NULL,
  C_W_ID       INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_079
INDEX IN is_customer_079
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 93601 ENDING AT
94800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER77.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER77;
CREATE TABLE CUSTOMER77
(
  C_ID          INTEGER    NOT NULL,
  C_STATE      CHAR(2)    NOT NULL,
  C_ZIP        CHAR(9)    NOT NULL,
  C_PHONE      CHAR(16)   NOT NULL,
  C_SINCE      TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)    NOT NULL,
  C_CREDIT     CHAR(2)    NOT NULL,
  C_DISCOUNT  REAL       NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16) NOT NULL,
  C_FIRST      VARCHAR(16) NOT NULL,
  C_STREET_1   VARCHAR(20) NOT NULL,
  C_STREET_2   VARCHAR(20) NOT NULL,
  C_CITY       VARCHAR(20) NOT NULL,
  C_D_ID       SMALLINT   NOT NULL,
  C_W_ID       INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,

```

```

  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_007
INDEX IN is_customer_007
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 7201 ENDING AT
8400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_CUSTOMER80.ddl

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER80;
CREATE TABLE CUSTOMER80
(
  C_ID          INTEGER    NOT NULL,
  C_STATE      CHAR(2)    NOT NULL,
  C_ZIP        CHAR(9)    NOT NULL,
  C_PHONE      CHAR(16)   NOT NULL,
  C_SINCE      TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)    NOT NULL,
  C_CREDIT     CHAR(2)    NOT NULL,
  C_DISCOUNT  REAL       NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16) NOT NULL,
  C_FIRST      VARCHAR(16) NOT NULL,
  C_STREET_1   VARCHAR(20) NOT NULL,
  C_STREET_2   VARCHAR(20) NOT NULL,
  C_CITY       VARCHAR(20) NOT NULL,
  C_D_ID       SMALLINT   NOT NULL,
  C_W_ID       INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT
NULL,
  C_PAYMENT_CNT INTEGER  NOT NULL
)
IN ts_customer_080
INDEX IN is_customer_080
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 94801 ENDING AT
96000,
  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 8401 ENDING AT
9600,
  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 9601 ENDING AT
10800,
  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 10801 ENDING AT
12000
)
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 12001 ENDING AT
13200
)
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 13201 ENDING AT
14400
)
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 14401 ENDING AT
15600
)
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 15601 ENDING AT
16800
)
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 16801 ENDING AT
18000
)
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 18001 ENDING AT
19200
)
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 19201 ENDING AT
20400
)
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 20401 ENDING AT

```

```

21600
)
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 21601 ENDING AT

```

```

22800
)
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 1200
)

```

```

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 22801 ENDING AT

```

```

24000
)
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 24001 ENDING AT

```

```

25200
)
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 25201 ENDING AT
26400
)
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 26401 ENDING AT
27600
)
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 27601 ENDING AT
28800
)
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 28801 ENDING AT
30000
)
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 30001 ENDING AT
31200
)
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 31201 ENDING AT
32400
    )
    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 32401 ENDING AT
33600
    )
    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 33601 ENDING AT
34800
    )
    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 1201 ENDING AT
2400
    )
    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 34801 ENDING AT
36000
    )
    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 36001 ENDING AT
37200
    )
    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 37201 ENDING AT
```

38400

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT33.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT33;
CREATE TABLE DISTRICT33

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_033
INDEX IN ts_dis_033
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
```

D_W_ID STARTING FROM 38401 ENDING AT
39600

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT34.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT34;
CREATE TABLE DISTRICT34

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
```

```
IN ts_dis_034
INDEX IN ts_dis_034
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 39601 ENDING AT
```

40800

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT35.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT35;
CREATE TABLE DISTRICT35

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
```

```
D_W_ID INTEGER NOT NULL
)
IN ts_dis_035
INDEX IN ts_dis_035
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 40801 ENDING AT
```

42000

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT36.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT36;
CREATE TABLE DISTRICT36

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
```

```
IN ts_dis_036
INDEX IN ts_dis_036
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 42001 ENDING AT
```

43200

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT37.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT37;
CREATE TABLE DISTRICT37

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
```

```

D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_037
INDEX IN ts_dis_037
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 43201 ENDING AT
44400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT38.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT38;
CREATE TABLE DISTRICT38
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_038
INDEX IN ts_dis_038
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 44401 ENDING AT
45600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT39.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT39;

```

```

CREATE TABLE DISTRICT39
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_039
INDEX IN ts_dis_039
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 45601 ENDING AT
46800
)
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 2401 ENDING AT
3600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT40.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT40;
CREATE TABLE DISTRICT40
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_040
INDEX IN ts_dis_040
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 46801 ENDING AT
48000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT41.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT41;
CREATE TABLE DISTRICT41
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_041
INDEX IN ts_dis_041
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,

```

```

D_W_ID STARTING FROM 48001 ENDING AT
49200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT42.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT42;
CREATE TABLE DISTRICT42
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_042
INDEX IN ts_dis_042
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 49201 ENDING AT
50400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT43.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT43;
CREATE TABLE DISTRICT43
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,

```

```

D_W_ID INTEGER NOT NULL
)
IN ts_dis_043
INDEX IN ts_dis_043
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 50401 ENDING AT
51600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT44.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT44;
CREATE TABLE DISTRICT44
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_044
INDEX IN ts_dis_044
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 51601 ENDING AT
52800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT45.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT45;
CREATE TABLE DISTRICT45
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,

```

```

D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_045
INDEX IN ts_dis_045
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 52801 ENDING AT
54000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT46.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT46;
CREATE TABLE DISTRICT46
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_046
INDEX IN ts_dis_046
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 54001 ENDING AT
55200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT47.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT47;

```

```

CREATE TABLE DISTRICT47
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_047
INDEX IN ts_dis_047
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 55201 ENDING AT
56400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT48.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT48;
CREATE TABLE DISTRICT48
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_048
INDEX IN ts_dis_048
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 56401 ENDING AT
57600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT49.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT49;
CREATE TABLE DISTRICT49
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_049
INDEX IN ts_dis_049
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 57601 ENDING AT
58800
)
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 3601 ENDING AT
4800
)
ALLOW OVERFLOW;
connect reset;

CRTB_DISTRICT50.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT50;
CREATE TABLE DISTRICT50
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_050
INDEX IN ts_dis_050
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 58801 ENDING AT
60000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT51.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT51;
CREATE TABLE DISTRICT51
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,

```

```

        D_W_ID INTEGER NOT NULL
    )
    IN ts_dis_051
    INDEX IN ts_dis_051
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 60001 ENDING AT
61200
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT52.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT52;
CREATE TABLE DISTRICT52

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,
    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_052
    INDEX IN ts_dis_052
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 61201 ENDING AT
62400
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT53.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT53;
CREATE TABLE DISTRICT53

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,

```

```

    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_053
    INDEX IN ts_dis_053
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 62401 ENDING AT
63600
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT54.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT54;
CREATE TABLE DISTRICT54

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,
    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_054
    INDEX IN ts_dis_054
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 63601 ENDING AT
64800
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT55.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT55;

```

```

CREATE TABLE DISTRICT55
(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,
    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_055
    INDEX IN ts_dis_055
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 64801 ENDING AT
66000
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT56.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT56;
CREATE TABLE DISTRICT56

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,
    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_056
    INDEX IN ts_dis_056
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 66001 ENDING AT
67200
    )
    ALLOW OVERFLOW;
connect reset;

```


CRTB_DISTRICT57.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT57;
CREATE TABLE DISTRICT57

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_057
INDEX IN ts_dis_057
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 67201 ENDING AT
```

68400

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT58.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT58;
CREATE TABLE DISTRICT58

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
```

```
IN ts_dis_058
INDEX IN ts_dis_058
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
```

D_W_ID STARTING FROM 68401 ENDING AT
69600

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT59.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT59;
CREATE TABLE DISTRICT59

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
```

```
IN ts_dis_059
INDEX IN ts_dis_059
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 69601 ENDING AT
```

70800

```
)
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 4801 ENDING AT
```

6000

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT60.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT60;
CREATE TABLE DISTRICT60

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
```

```
IN ts_dis_060
INDEX IN ts_dis_060
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 70801 ENDING AT
```

72000

```
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_DISTRICT61.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT61;
CREATE TABLE DISTRICT61

```
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
```

```

D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_061
INDEX IN ts_dis_061
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 72001 ENDING AT

```

73200

```

)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT62.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT62;
CREATE TABLE DISTRICT62

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_062
INDEX IN ts_dis_062
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 73201 ENDING AT

```

74400

```

)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT63.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT63;

```

```

CREATE TABLE DISTRICT63
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_063
INDEX IN ts_dis_063
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 74401 ENDING AT

```

75600

```

)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT64.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT64;
CREATE TABLE DISTRICT64

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_064
INDEX IN ts_dis_064
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 75601 ENDING AT

```

76800

```

)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT65.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT65;
CREATE TABLE DISTRICT65

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_065
INDEX IN ts_dis_065
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 76801 ENDING AT

```

78000

```

)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT66.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT66;
CREATE TABLE DISTRICT66

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_066
INDEX IN ts_dis_066
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,

```

```

D_W_ID STARTING FROM 78001 ENDING AT
79200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT67.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT67;
CREATE TABLE DISTRICT67
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_067
INDEX IN ts_dis_067
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 79201 ENDING AT
80400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT68.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT68;
CREATE TABLE DISTRICT68
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,

```

```

D_W_ID INTEGER NOT NULL
)
IN ts_dis_068
INDEX IN ts_dis_068
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 80401 ENDING AT
81600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT69.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT69;
CREATE TABLE DISTRICT69
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_069
INDEX IN ts_dis_069
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 81601 ENDING AT
82800
)
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 6001 ENDING AT
7200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT70.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT70;
CREATE TABLE DISTRICT70
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dis_070
INDEX IN ts_dis_070
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 82801 ENDING AT
84000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT71.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT71;

```

```

CREATE TABLE DISTRICT71
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_071
INDEX IN ts_dis_071
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 84001 ENDING AT
85200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT72.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT72;
CREATE TABLE DISTRICT72
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_072
INDEX IN ts_dis_072
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 85201 ENDING AT
86400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT73.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT73;
CREATE TABLE DISTRICT73
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_073
INDEX IN ts_dis_073
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 86401 ENDING AT
87600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT74.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT74;
CREATE TABLE DISTRICT74
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_074
INDEX IN ts_dis_074
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,

```

```

D_W_ID STARTING FROM 87601 ENDING AT
88800
)
ALLOW OVERFLOW;
connect reset;

CRTB_DISTRICT75.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT75;
CREATE TABLE DISTRICT75
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dis_075
INDEX IN ts_dis_075
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 88801 ENDING AT
90000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT76.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT76;
CREATE TABLE DISTRICT76
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,

```

```

        D_W_ID INTEGER NOT NULL
    )
    IN ts_dis_076
    INDEX IN ts_dis_076
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 90001 ENDING AT
91200
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT77.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT77;
CREATE TABLE DISTRICT77

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,
    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_077
    INDEX IN ts_dis_077
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 91201 ENDING AT
92400
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT78.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT78;
CREATE TABLE DISTRICT78

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,

```

```

    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_078
    INDEX IN ts_dis_078
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 92401 ENDING AT
93600
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT79.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT79;
CREATE TABLE DISTRICT79

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,
    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_079
    INDEX IN ts_dis_079
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 93601 ENDING AT
94800
    )
    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 7201 ENDING AT
8400
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_DISTRICT80.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT80;
CREATE TABLE DISTRICT80

```

```

(
    D_NEXT_O_ID INTEGER NOT NULL,
    D_TAX REAL NOT NULL,
    D_YTD DECIMAL(12,2) NOT NULL,
    D_NAME CHAR(10) NOT NULL,
    D_STREET_1 CHAR(20) NOT NULL,
    D_STREET_2 CHAR(20) NOT NULL,
    D_CITY CHAR(20) NOT NULL,
    D_STATE CHAR(2) NOT NULL,
    D_ZIP CHAR(9) NOT NULL,
    D_ID SMALLINT NOT NULL,
    D_W_ID INTEGER NOT NULL
)
    IN ts_dis_080
    INDEX IN ts_dis_080
    ORGANIZE BY KEY SEQUENCE (
        D_ID STARTING FROM 1 ENDING AT 10,
        D_W_ID STARTING FROM 94801 ENDING AT
96000
    )
    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 8401 ENDING AT
```

9600

```
)
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 9601 ENDING AT
10800
```

```
)
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_HISTORY33.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY33;
CREATE TABLE HISTORY33

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_033
    INDEX IN ts_history_033;
ALTER TABLE HISTORY33 APPEND ON;
connect reset;

```

CRTB_HISTORY34.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY34;
CREATE TABLE HISTORY34

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_034
    INDEX IN ts_history_034;
ALTER TABLE HISTORY34 APPEND ON;
connect reset;

```

CRTB_HISTORY35.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY35;
CREATE TABLE HISTORY35
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_035
INDEX IN ts_history_035;
ALTER TABLE HISTORY35 APPEND ON;
connect reset;

```

CRTB_HISTORY36.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY36;
CREATE TABLE HISTORY36
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_036
INDEX IN ts_history_036;
ALTER TABLE HISTORY36 APPEND ON;
connect reset;

```

CRTB_HISTORY37.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY37;
CREATE TABLE HISTORY37
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,

```

```

  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_037
INDEX IN ts_history_037;
ALTER TABLE HISTORY37 APPEND ON;
connect reset;

```

CRTB_HISTORY38.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY38;
CREATE TABLE HISTORY38
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_038
INDEX IN ts_history_038;
ALTER TABLE HISTORY38 APPEND ON;
connect reset;

```

CRTB_HISTORY39.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY39;
CREATE TABLE HISTORY39
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_039
INDEX IN ts_history_039;
ALTER TABLE HISTORY39 APPEND ON;
connect reset;

```

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_HISTORY40.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY40;
CREATE TABLE HISTORY40
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_040
INDEX IN ts_history_040;
ALTER TABLE HISTORY40 APPEND ON;
connect reset;

```

CRTB_HISTORY41.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY41;
CREATE TABLE HISTORY41
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,

```

```

        H_AMOUNT    DECIMAL(6,2) NOT NULL,
        H_DATA      CHAR(24)   NOT NULL
    )
    IN ts_history_041
    INDEX IN ts_history_041;
ALTER TABLE HISTORY41 APPEND ON;
connect reset;

```

CRTB_HISTORY42.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY42;
CREATE TABLE HISTORY42

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_042
    INDEX IN ts_history_042;
ALTER TABLE HISTORY42 APPEND ON;
connect reset;

```

CRTB_HISTORY43.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY43;
CREATE TABLE HISTORY43

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_043
    INDEX IN ts_history_043;
ALTER TABLE HISTORY43 APPEND ON;
connect reset;

```

CRTB_HISTORY44.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY44;
CREATE TABLE HISTORY44

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_044
    INDEX IN ts_history_044;
ALTER TABLE HISTORY44 APPEND ON;
connect reset;

```

CRTB_HISTORY45.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY45;
CREATE TABLE HISTORY45

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_045
    INDEX IN ts_history_045;
ALTER TABLE HISTORY45 APPEND ON;
connect reset;

```

CRTB_HISTORY46.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY46;
CREATE TABLE HISTORY46

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,

```

```

        H_AMOUNT    DECIMAL(6,2) NOT NULL,
        H_DATA      CHAR(24)   NOT NULL
    )

```

```

    IN ts_history_046
    INDEX IN ts_history_046;
ALTER TABLE HISTORY46 APPEND ON;
connect reset;

```

CRTB_HISTORY47.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY47;
CREATE TABLE HISTORY47

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_047
    INDEX IN ts_history_047;
ALTER TABLE HISTORY47 APPEND ON;
connect reset;

```

CRTB_HISTORY48.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY48;
CREATE TABLE HISTORY48

```

```

(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP  NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_048
    INDEX IN ts_history_048;
ALTER TABLE HISTORY48 APPEND ON;
connect reset;

```

CRTB_HISTORY49.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY49;
CREATE TABLE HISTORY49
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT   NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT   NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP  NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_049
INDEX IN ts_history_049;
ALTER TABLE HISTORY49 APPEND ON;
connect reset;

```

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_HISTORY50.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY50;
CREATE TABLE HISTORY50
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT   NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT   NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP  NOT NULL,

```

```

  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_050
INDEX IN ts_history_050;
ALTER TABLE HISTORY50 APPEND ON;
connect reset;

```

CRTB_HISTORY51.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY51;
CREATE TABLE HISTORY51
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT   NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT   NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP  NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_051
INDEX IN ts_history_051;
ALTER TABLE HISTORY51 APPEND ON;
connect reset;

```

CRTB_HISTORY52.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY52;
CREATE TABLE HISTORY52
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT   NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT   NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP  NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_052
INDEX IN ts_history_052;
ALTER TABLE HISTORY52 APPEND ON;
connect reset;

```

CRTB_HISTORY53.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY53;
CREATE TABLE HISTORY53
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT   NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT   NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP  NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_053
INDEX IN ts_history_053;
ALTER TABLE HISTORY53 APPEND ON;
connect reset;

```

CRTB_HISTORY54.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY54;
CREATE TABLE HISTORY54
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT   NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT   NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP  NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_054
INDEX IN ts_history_054;
ALTER TABLE HISTORY54 APPEND ON;
connect reset;

```

CRTB_HISTORY55.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY55;
CREATE TABLE HISTORY55
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT   NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT   NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP  NOT NULL,

```

```

        H_AMOUNT    DECIMAL(6,2) NOT NULL,
        H_DATA      CHAR(24)   NOT NULL
    )
    IN ts_history_055
    INDEX IN ts_history_055;
ALTER TABLE HISTORY55 APPEND ON;
connect reset;

```

CRTB_HISTORY56.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY56;
CREATE TABLE HISTORY56
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_056
INDEX IN ts_history_056;
ALTER TABLE HISTORY56 APPEND ON;
connect reset;

```

CRTB_HISTORY57.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY57;
CREATE TABLE HISTORY57
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_057
INDEX IN ts_history_057;
ALTER TABLE HISTORY57 APPEND ON;
connect reset;

```

CRTB_HISTORY58.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY58;
CREATE TABLE HISTORY58
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_058
INDEX IN ts_history_058;
ALTER TABLE HISTORY58 APPEND ON;
connect reset;

```

CRTB_HISTORY59.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY59;
CREATE TABLE HISTORY59
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_059
INDEX IN ts_history_059;
ALTER TABLE HISTORY59 APPEND ON;
connect reset;

```

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_HISTORY60.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY60;
CREATE TABLE HISTORY60
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_060
INDEX IN ts_history_060;
ALTER TABLE HISTORY60 APPEND ON;
connect reset;

```

CRTB_HISTORY61.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY61;
CREATE TABLE HISTORY61
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT  NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT  NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    TIMESTAMP NOT NULL,
    H_AMOUNT  DECIMAL(6,2) NOT NULL,
    H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_061
INDEX IN ts_history_061;
ALTER TABLE HISTORY61 APPEND ON;
connect reset;

```

CRTB_HISTORY62.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY62;
CREATE TABLE HISTORY62
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_062
INDEX IN ts_history_062;
ALTER TABLE HISTORY62 APPEND ON;
connect reset;

```

CRTB_HISTORY63.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY63;
CREATE TABLE HISTORY63
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_063
INDEX IN ts_history_063;
ALTER TABLE HISTORY63 APPEND ON;
connect reset;

```

CRTB_HISTORY64.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY64;
CREATE TABLE HISTORY64
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,

```

```

  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_064
INDEX IN ts_history_064;
ALTER TABLE HISTORY64 APPEND ON;
connect reset;

```

CRTB_HISTORY65.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY65;
CREATE TABLE HISTORY65
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_065
INDEX IN ts_history_065;
ALTER TABLE HISTORY65 APPEND ON;
connect reset;

```

CRTB_HISTORY66.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY66;
CREATE TABLE HISTORY66
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_066
INDEX IN ts_history_066;
ALTER TABLE HISTORY66 APPEND ON;
connect reset;

```

CRTB_HISTORY67.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY67;
CREATE TABLE HISTORY67
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_067
INDEX IN ts_history_067;
ALTER TABLE HISTORY67 APPEND ON;
connect reset;

```

CRTB_HISTORY68.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY68;
CREATE TABLE HISTORY68
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,
  H_AMOUNT    DECIMAL(6,2) NOT NULL,
  H_DATA      CHAR(24)   NOT NULL
)
IN ts_history_068
INDEX IN ts_history_068;
ALTER TABLE HISTORY68 APPEND ON;
connect reset;

```

CRTB_HISTORY69.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY69;
CREATE TABLE HISTORY69
(
  H_C_ID      INTEGER    NOT NULL,
  H_C_D_ID    SMALLINT  NOT NULL,
  H_C_W_ID    INTEGER    NOT NULL,
  H_D_ID      SMALLINT  NOT NULL,
  H_W_ID      INTEGER    NOT NULL,
  H_DATE      TIMESTAMP NOT NULL,

```

```

        H_AMOUNT    DECIMAL(6,2) NOT NULL,
        H_DATA      CHAR(24)   NOT NULL
    )
    IN ts_history_069
    INDEX IN ts_history_069;
ALTER TABLE HISTORY69 APPEND ON;
connect reset;

```

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_HISTORY70.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY70;
CREATE TABLE HISTORY70

```

```

(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_070
    INDEX IN ts_history_070;
ALTER TABLE HISTORY70 APPEND ON;
connect reset;

```

CRTB_HISTORY71.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY71;
CREATE TABLE HISTORY71

```

```

(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_071
    INDEX IN ts_history_071;
ALTER TABLE HISTORY71 APPEND ON;
connect reset;

```

CRTB_HISTORY72.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY72;
CREATE TABLE HISTORY72

```

```

(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_072
    INDEX IN ts_history_072;
ALTER TABLE HISTORY72 APPEND ON;
connect reset;

```

CRTB_HISTORY73.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY73;
CREATE TABLE HISTORY73

```

```

(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,

```

```

        H_AMOUNT    DECIMAL(6,2) NOT NULL,
        H_DATA      CHAR(24)   NOT NULL
    )
    IN ts_history_073
    INDEX IN ts_history_073;
ALTER TABLE HISTORY73 APPEND ON;
connect reset;

```

CRTB_HISTORY74.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY74;
CREATE TABLE HISTORY74

```

```

(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_074
    INDEX IN ts_history_074;
ALTER TABLE HISTORY74 APPEND ON;
connect reset;

```

CRTB_HISTORY75.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY75;
CREATE TABLE HISTORY75

```

```

(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)

```

```

    IN ts_history_075
    INDEX IN ts_history_075;
ALTER TABLE HISTORY75 APPEND ON;
connect reset;

```

CRTB_HISTORY76.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY76;
CREATE TABLE HISTORY76
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_076
INDEX IN ts_history_076;
ALTER TABLE HISTORY76 APPEND ON;
connect reset;

```

CRTB_HISTORY77.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY77;
CREATE TABLE HISTORY77
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_077
INDEX IN ts_history_077;
ALTER TABLE HISTORY77 APPEND ON;
connect reset;

```

CRTB_HISTORY78.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY78;
CREATE TABLE HISTORY78
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,

```

```

  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_078
INDEX IN ts_history_078;
ALTER TABLE HISTORY78 APPEND ON;
connect reset;

```

CRTB_HISTORY79.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY79;
CREATE TABLE HISTORY79
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_079
INDEX IN ts_history_079;
ALTER TABLE HISTORY79 APPEND ON;
connect reset;

```

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_HISTORY80.ddl

```

connect to TPCC in share mode;
DROP TABLE HISTORY80;
CREATE TABLE HISTORY80
(
  H_C_ID    INTEGER    NOT NULL,
  H_C_D_ID  SMALLINT   NOT NULL,
  H_C_W_ID  INTEGER    NOT NULL,
  H_D_ID    SMALLINT   NOT NULL,
  H_W_ID    INTEGER    NOT NULL,
  H_DATE    TIMESTAMP  NOT NULL,
  H_AMOUNT  DECIMAL(6,2) NOT NULL,
  H_DATA    CHAR(24)   NOT NULL
)
IN ts_history_080
INDEX IN ts_history_080;
ALTER TABLE HISTORY80 APPEND ON;
connect reset;

```

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 10801 ENDING
AT 12000,
    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 12001 ENDING
AT 13200,
    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 13201 ENDING
AT 14400,
    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 14401 ENDING
AT 15600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

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 15601 ENDING
AT 16800,
    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 16801 ENDING
AT 18000,
        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 18001 ENDING
AT 19200,
    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 19201 ENDING
AT 20400,
    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 20401 ENDING
AT 21600,
    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 21601 ENDING
AT 22800,
    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
1200,
    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 22801 ENDING
AT 24000,
    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 24001 ENDING
AT 25200,
        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 25201 ENDING
AT 26400,
    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 26401 ENDING
AT 27600,
        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 27601 ENDING
AT 28800,
    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 28801 ENDING
AT 30000,
    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 30001 ENDING
AT 31200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

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 31201 ENDING
AT 32400,
    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 32401 ENDING
AT 33600,
  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 33601 ENDING
AT 34800,
  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 1201 ENDING AT
2400,
  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 34801 ENDING
AT 36000,
  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 36001 ENDING
AT 37200,
  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 37201 ENDING
AT 38400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA33.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA33;
CREATE TABLE NEW_ORDERA33
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_033
INDEX IN ts_newordA_033
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 38401 ENDING
AT 39600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)

```

```
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA34.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA34;
CREATE TABLE NEW_ORDERA34
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_034
INDEX IN ts_newordA_034
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 39601 ENDING
```

```
AT 40800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA35.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA35;
CREATE TABLE NEW_ORDERA35
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_035
INDEX IN ts_newordA_035
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 40801 ENDING
```

```
AT 42000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA36.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA36;
CREATE TABLE NEW_ORDERA36
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_036
INDEX IN ts_newordA_036
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 42001 ENDING
```

```
AT 43200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA37.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA37;
CREATE TABLE NEW_ORDERA37
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_037
INDEX IN ts_newordA_037
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 43201 ENDING
```

```
AT 44400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA38.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA38;
CREATE TABLE NEW_ORDERA38
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
```

```
NO_W_ID    INTEGER    NOT NULL
```

```
)
IN ts_newordA_038
INDEX IN ts_newordA_038
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 44401 ENDING
```

```
AT 45600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA39.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA39;
CREATE TABLE NEW_ORDERA39
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_039
INDEX IN ts_newordA_039
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 45601 ENDING
```

```
AT 46800,
  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 (
```

```

3600, NO_W_ID STARTING FROM 2401 ENDING AT
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA40.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA40;
CREATE TABLE NEW_ORDERA40
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_040
INDEX IN ts_newordA_040
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 46801 ENDING
AT 48000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA41.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA41;
CREATE TABLE NEW_ORDERA41
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_041
INDEX IN ts_newordA_041
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 48001 ENDING
AT 49200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)

```

```

ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA42.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA42;
CREATE TABLE NEW_ORDERA42
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_042
INDEX IN ts_newordA_042
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 49201 ENDING
AT 50400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA43.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA43;
CREATE TABLE NEW_ORDERA43
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_043
INDEX IN ts_newordA_043
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 50401 ENDING
AT 51600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA44.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA44;
CREATE TABLE NEW_ORDERA44
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_044
INDEX IN ts_newordA_044
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 51601 ENDING
AT 52800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA45.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA45;
CREATE TABLE NEW_ORDERA45
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_045
INDEX IN ts_newordA_045
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 52801 ENDING
AT 54000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA46.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA46;
CREATE TABLE NEW_ORDERA46
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,

```

```

        NO_W_ID    INTEGER    NOT NULL
    )
    IN ts_newordA_046
    INDEX IN ts_newordA_046
    ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 54001 ENDING
AT 55200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA47.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA47;
CREATE TABLE NEW_ORDERA47
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_047
INDEX IN ts_newordA_047
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 55201 ENDING
AT 56400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA48.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA48;
CREATE TABLE NEW_ORDERA48
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_048
INDEX IN ts_newordA_048
ORGANIZE BY KEY SEQUENCE (

```

```

        NO_W_ID STARTING FROM 56401 ENDING
AT 57600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA49.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA49;
CREATE TABLE NEW_ORDERA49
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_049
INDEX IN ts_newordA_049
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 57601 ENDING
AT 58800,
    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 3601 ENDING AT
4800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)

```

```

        ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA50.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA50;
CREATE TABLE NEW_ORDERA50
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_050
INDEX IN ts_newordA_050
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 58801 ENDING
AT 60000,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA51.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA51;
CREATE TABLE NEW_ORDERA51
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_051
INDEX IN ts_newordA_051
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 60001 ENDING
AT 61200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA52.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA52;
CREATE TABLE NEW_ORDERA52
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_052
INDEX IN ts_newordA_052
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 61201 ENDING
AT 62400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA53.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA53;
CREATE TABLE NEW_ORDERA53
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_053
INDEX IN ts_newordA_053
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 62401 ENDING
AT 63600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA54.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA54;
CREATE TABLE NEW_ORDERA54
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,

```

```

  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_054
INDEX IN ts_newordA_054
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 63601 ENDING
AT 64800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA55.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA55;
CREATE TABLE NEW_ORDERA55
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_055
INDEX IN ts_newordA_055
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 64801 ENDING
AT 66000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA56.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA56;
CREATE TABLE NEW_ORDERA56
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_056
INDEX IN ts_newordA_056
ORGANIZE BY KEY SEQUENCE (

```

```

  NO_W_ID STARTING FROM 66001 ENDING
AT 67200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA57.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA57;
CREATE TABLE NEW_ORDERA57
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_057
INDEX IN ts_newordA_057
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 67201 ENDING
AT 68400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA58.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA58;
CREATE TABLE NEW_ORDERA58
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_058
INDEX IN ts_newordA_058
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 68401 ENDING
AT 69600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)

```



```
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERA59.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA59;
CREATE TABLE NEW_ORDERA59
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_059
INDEX IN ts_newordA_059
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 69601 ENDING
```

```
AT 70800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA55.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA55;
CREATE TABLE NEW_ORDERA55
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_005
INDEX IN ts_newordA_005
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 4801 ENDING AT
```

```
6000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA60.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA60;
CREATE TABLE NEW_ORDERA60
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_060
INDEX IN ts_newordA_060
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 70801 ENDING
```

```
AT 72000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA61.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA61;
CREATE TABLE NEW_ORDERA61
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_061
INDEX IN ts_newordA_061
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 72001 ENDING
```

```
AT 73200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA62.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA62;
CREATE TABLE NEW_ORDERA62
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
```

```
NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_062
INDEX IN ts_newordA_062
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 73201 ENDING
```

```
AT 74400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA63.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA63;
CREATE TABLE NEW_ORDERA63
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_063
INDEX IN ts_newordA_063
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 74401 ENDING
```

```
AT 75600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
```

```
3675
```

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERA64.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA64;
CREATE TABLE NEW_ORDERA64
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordA_064
INDEX IN ts_newordA_064
ORGANIZE BY KEY SEQUENCE (
```

```

NO_W_ID STARTING FROM 75601 ENDING
AT 76800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA65.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA65;
CREATE TABLE NEW_ORDERA65
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_065
INDEX IN ts_newordA_065
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 76801 ENDING
AT 78000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA66.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA66;
CREATE TABLE NEW_ORDERA66
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_066
INDEX IN ts_newordA_066
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 78001 ENDING
AT 79200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)

```

```

ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA67.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA67;
CREATE TABLE NEW_ORDERA67
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_067
INDEX IN ts_newordA_067
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 79201 ENDING
AT 80400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA68.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA68;
CREATE TABLE NEW_ORDERA68
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_068
INDEX IN ts_newordA_068
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 80401 ENDING
AT 81600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA69.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA69;
CREATE TABLE NEW_ORDERA69
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_069
INDEX IN ts_newordA_069
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 81601 ENDING
AT 82800,
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 6001 ENDING AT
7200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA70.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA70;
CREATE TABLE NEW_ORDERA70
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,

```

```

        NO_W_ID    INTEGER    NOT NULL
    )
    IN ts_newordA_070
    INDEX IN ts_newordA_070
    ORGANIZE BY KEY SEQUENCE (
        NO_W_ID STARTING FROM 82801 ENDING
AT 84000,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 1900 ENDING AT
3675
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA71.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA71;
CREATE TABLE NEW_ORDERA71
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_071
INDEX IN ts_newordA_071
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 84001 ENDING
AT 85200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA72.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA72;
CREATE TABLE NEW_ORDERA72
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_072
INDEX IN ts_newordA_072
ORGANIZE BY KEY SEQUENCE (

```

```

        NO_W_ID STARTING FROM 85201 ENDING
AT 86400,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 1900 ENDING AT
3675
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA73.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA73;
CREATE TABLE NEW_ORDERA73
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_073
INDEX IN ts_newordA_073
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 86401 ENDING
AT 87600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA74.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA74;
CREATE TABLE NEW_ORDERA74
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_074
INDEX IN ts_newordA_074
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 87601 ENDING
AT 88800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)

```

```

        ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA75.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA75;
CREATE TABLE NEW_ORDERA75
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_075
INDEX IN ts_newordA_075
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 88801 ENDING
AT 90000,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA76.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA76;
CREATE TABLE NEW_ORDERA76
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_076
INDEX IN ts_newordA_076
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 90001 ENDING
AT 91200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA77.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA77;
CREATE TABLE NEW_ORDERA77
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_077
INDEX IN ts_newordA_077
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 91201 ENDING
AT 92400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA78.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA78;
CREATE TABLE NEW_ORDERA78
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_078
INDEX IN ts_newordA_078
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 92401 ENDING
AT 93600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA79.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA79;
CREATE TABLE NEW_ORDERA79
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,

```

```

  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_079
INDEX IN ts_newordA_079
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 93601 ENDING
AT 94800,
  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 7201 ENDING AT
8400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT
3675
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERA80.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA80;
CREATE TABLE NEW_ORDERA80
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_080
INDEX IN ts_newordA_080
ORGANIZE BY KEY SEQUENCE (

```

```

  NO_W_ID STARTING FROM 94801 ENDING
AT 96000,
  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 8401 ENDING AT
9600,
  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 9601 ENDING AT
10800,
  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 10801 ENDING
```

AT 12000,

```
  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 12001 ENDING
```

AT 13200,

```
  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 13201 ENDING
```

AT 14400,

```
  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 14401 ENDING
```

AT 15600,

```
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
```

5451

```
)
ALLOW OVERFLOW;
```

connect reset;

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 15601 ENDING
```

AT 16800,

```
  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 16801 ENDING
```

AT 18000,

```
  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 18001 ENDING
AT 19200,
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 19201 ENDING
AT 20400,
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 20401 ENDING
AT 21600,
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 21601 ENDING
AT 22800,
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
1200,
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 22801 ENDING
AT 24000,
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 24001 ENDING
AT 25200,
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 25201 ENDING
AT 26400,
    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 26401 ENDING
AT 27600,
    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 27601 ENDING
AT 28800,
    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 28801 ENDING
AT 30000,
    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 30001 ENDING
AT 31200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)

```

```

        ALLOW OVERFLOW;
connect reset;

```

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 31201 ENDING
AT 32400,
    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 32401 ENDING
AT 33600,
    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 33601 ENDING
AT 34800,
    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 1201 ENDING AT
2400,
    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 34801 ENDING
AT 36000,
    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 36001 ENDING
AT 37200,
    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 37201 ENDING
AT 38400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB33.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB33;
CREATE TABLE NEW_ORDERB33
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_033
INDEX IN ts_newordB_033
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 38401 ENDING
AT 39600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB34.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB34;
CREATE TABLE NEW_ORDERB34
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_034
INDEX IN ts_newordB_034
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 39601 ENDING
AT 40800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)

```



```
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB35.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB35;
CREATE TABLE NEW_ORDERB35
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_035
INDEX IN ts_newordB_035
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 40801 ENDING
AT 42000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB36.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB36;
CREATE TABLE NEW_ORDERB36
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_036
INDEX IN ts_newordB_036
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 42001 ENDING
AT 43200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB37.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB37;
CREATE TABLE NEW_ORDERB37
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_037
INDEX IN ts_newordB_037
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 43201 ENDING
AT 44400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB38.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB38;
CREATE TABLE NEW_ORDERB38
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_038
INDEX IN ts_newordB_038
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 44401 ENDING
AT 45600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB39.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB39;
CREATE TABLE NEW_ORDERB39
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
```

```
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_039
INDEX IN ts_newordB_039
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 45601 ENDING
AT 46800,
  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 2401 ENDING AT
3600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_NEW_ORDERB40.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB40;
CREATE TABLE NEW_ORDERB40
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_040
INDEX IN ts_newordB_040
ORGANIZE BY KEY SEQUENCE (
```

```

NO_W_ID STARTING FROM 46801 ENDING
AT 48000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB41.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB41;
CREATE TABLE NEW_ORDERB41
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_041
INDEX IN ts_newordB_041
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 48001 ENDING
AT 49200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB42.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB42;
CREATE TABLE NEW_ORDERB42
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_042
INDEX IN ts_newordB_042
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 49201 ENDING
AT 50400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)

```

```

ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB43.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB43;
CREATE TABLE NEW_ORDERB43
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_043
INDEX IN ts_newordB_043
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 50401 ENDING
AT 51600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB44.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB44;
CREATE TABLE NEW_ORDERB44
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_044
INDEX IN ts_newordB_044
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 51601 ENDING
AT 52800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB45.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB45;
CREATE TABLE NEW_ORDERB45
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_045
INDEX IN ts_newordB_045
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 52801 ENDING
AT 54000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB46.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB46;
CREATE TABLE NEW_ORDERB46
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_046
INDEX IN ts_newordB_046
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 54001 ENDING
AT 55200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB47.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB47;
CREATE TABLE NEW_ORDERB47
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,

```

```

        NO_W_ID    INTEGER    NOT NULL
    )
    IN ts_newordB_047
    INDEX IN ts_newordB_047
    ORGANIZE BY KEY SEQUENCE (
        NO_W_ID STARTING FROM 55201 ENDING
AT 56400,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 3676 ENDING AT
5451
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB48.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB48;
CREATE TABLE NEW_ORDERB48
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_048
INDEX IN ts_newordB_048
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 56401 ENDING
AT 57600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB49.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB49;
CREATE TABLE NEW_ORDERB49
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_049
INDEX IN ts_newordB_049
ORGANIZE BY KEY SEQUENCE (

```

```

        NO_W_ID STARTING FROM 57601 ENDING
AT 58800,
        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 3601 ENDING AT
4800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB50.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB50;
CREATE TABLE NEW_ORDERB50
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_050
INDEX IN ts_newordB_050
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 58801 ENDING
AT 60000,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)

```

```

        ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB51.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB51;
CREATE TABLE NEW_ORDERB51
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_051
INDEX IN ts_newordB_051
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 60001 ENDING
AT 61200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB52.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB52;
CREATE TABLE NEW_ORDERB52
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_052
INDEX IN ts_newordB_052
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 61201 ENDING
AT 62400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB53.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB53;
CREATE TABLE NEW_ORDERB53
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_053
INDEX IN ts_newordB_053
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 62401 ENDING
AT 63600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB54.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB54;
CREATE TABLE NEW_ORDERB54
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_054
INDEX IN ts_newordB_054
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 63601 ENDING
AT 64800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB55.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB55;
CREATE TABLE NEW_ORDERB55
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,

```

```

    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_055
INDEX IN ts_newordB_055
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 64801 ENDING
AT 66000,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB56.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB56;
CREATE TABLE NEW_ORDERB56
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_056
INDEX IN ts_newordB_056
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 66001 ENDING
AT 67200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB57.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB57;
CREATE TABLE NEW_ORDERB57
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_057
INDEX IN ts_newordB_057
ORGANIZE BY KEY SEQUENCE (

```

```

    NO_W_ID STARTING FROM 67201 ENDING
AT 68400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB58.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB58;
CREATE TABLE NEW_ORDERB58
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_058
INDEX IN ts_newordB_058
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 68401 ENDING
AT 69600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB59.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB59;
CREATE TABLE NEW_ORDERB59
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_059
INDEX IN ts_newordB_059
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 69601 ENDING
AT 70800,
    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 4801 ENDING AT
```

6000,

```
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
```

5451

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERB60.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB60;
CREATE TABLE NEW_ORDERB60
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_060
INDEX IN ts_newordB_060
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 70801 ENDING
```

AT 72000,

```
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
```

5451

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERB61.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB61;
CREATE TABLE NEW_ORDERB61
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordB_061
INDEX IN ts_newordB_061
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 72001 ENDING
```

AT 73200,

```
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
```

5451

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERB62.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB62;
CREATE TABLE NEW_ORDERB62
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordB_062
INDEX IN ts_newordB_062
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 73201 ENDING
```

AT 74400,

```
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
```

5451

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERB63.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB63;
CREATE TABLE NEW_ORDERB63
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
```

```
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordB_063
INDEX IN ts_newordB_063
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 74401 ENDING
```

AT 75600,

```
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
```

5451

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERB64.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB64;
CREATE TABLE NEW_ORDERB64
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordB_064
INDEX IN ts_newordB_064
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 75601 ENDING
```

AT 76800,

```
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT
```

5451

```
)
ALLOW OVERFLOW;
```

```
connect reset;
```

CRTB_NEW_ORDERB65.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB65;
CREATE TABLE NEW_ORDERB65
```

```
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
```

```
IN ts_newordB_065
INDEX IN ts_newordB_065
ORGANIZE BY KEY SEQUENCE (
```

```

NO_W_ID STARTING FROM 76801 ENDING
AT 78000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB66.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB66;
CREATE TABLE NEW_ORDERB66
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_066
INDEX IN ts_newordB_066
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 78001 ENDING
AT 79200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB67.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB67;
CREATE TABLE NEW_ORDERB67
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_067
INDEX IN ts_newordB_067
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 79201 ENDING
AT 80400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)

```

```

ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB68.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB68;
CREATE TABLE NEW_ORDERB68
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_068
INDEX IN ts_newordB_068
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 80401 ENDING
AT 81600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB69.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB69;
CREATE TABLE NEW_ORDERB69
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_069
INDEX IN ts_newordB_069
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 81601 ENDING
AT 82800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB66.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 6001 ENDING AT
7200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB70.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB70;
CREATE TABLE NEW_ORDERB70
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_070
INDEX IN ts_newordB_070
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 82801 ENDING
AT 84000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB71.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB71;
CREATE TABLE NEW_ORDERB71
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,

```

```

        NO_W_ID    INTEGER    NOT NULL
    )
    IN ts_newordB_071
    INDEX IN ts_newordB_071
    ORGANIZE BY KEY SEQUENCE (
        NO_W_ID STARTING FROM 84001 ENDING
AT 85200,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 3676 ENDING AT
5451
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB72.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB72;
CREATE TABLE NEW_ORDERB72
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_072
INDEX IN ts_newordB_072
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 85201 ENDING
AT 86400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB73.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB73;
CREATE TABLE NEW_ORDERB73
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_073
INDEX IN ts_newordB_073
ORGANIZE BY KEY SEQUENCE (

```

```

        NO_W_ID STARTING FROM 86401 ENDING
AT 87600,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 3676 ENDING AT
5451
    )
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB74.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB74;
CREATE TABLE NEW_ORDERB74
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_074
INDEX IN ts_newordB_074
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 87601 ENDING
AT 88800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB75.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB75;
CREATE TABLE NEW_ORDERB75
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_075
INDEX IN ts_newordB_075
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 88801 ENDING
AT 90000,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)

```

```

        ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB76.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB76;
CREATE TABLE NEW_ORDERB76
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_076
INDEX IN ts_newordB_076
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 90001 ENDING
AT 91200,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB77.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB77;
CREATE TABLE NEW_ORDERB77
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_077
INDEX IN ts_newordB_077
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 91201 ENDING
AT 92400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
    ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB78.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB78;
CREATE TABLE NEW_ORDERB78
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_078
INDEX IN ts_newordB_078
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 92401 ENDING
AT 93600,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB79.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB79;
CREATE TABLE NEW_ORDERB79
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_079
INDEX IN ts_newordB_079
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 93601 ENDING
AT 94800,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB77.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 7201 ENDING AT
8400,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3676 ENDING AT
5451
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_NEW_ORDERB80.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB80;
CREATE TABLE NEW_ORDERB80
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT   NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_080
INDEX IN ts_newordB_080
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 94801 ENDING
AT 96000,
    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 8401 ENDING AT
9600,
    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 9601 ENDING AT
10800,
    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 10801 ENDING AT
12000,
  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 12001 ENDING AT
13200,
  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 13201 ENDING AT
14400,
  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 14401 ENDING AT
15600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT
3675,
  OL_NUMBER STARTING FROM 1 ENDING AT
15

```

```

)
ALLOW OVERFLOW;
connect reset;

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 15601 ENDING AT
16800,
  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 16801 ENDING AT
18000,
        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 18001 ENDING AT
19200,
        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 19201 ENDING AT
20400,
        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 20401 ENDING AT
21600,
        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 21601 ENDING AT
22800,
        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
1200,
  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 22801 ENDING AT
24000,
  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 24001 ENDING AT
25200,
  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 25201 ENDING AT
26400,
  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 26401 ENDING AT
27600,
  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 27601 ENDING AT
28800,
  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 28801 ENDING AT
30000,
  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 30001 ENDING AT
31200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT
3675,
  OL_NUMBER STARTING FROM 1 ENDING AT
15
)
ALLOW OVERFLOW;
connect reset;

```

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 31201 ENDING AT
32400,
  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 32401 ENDING AT
33600,
  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 33601 ENDING AT
34800,
  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 1201 ENDING AT
2400,
  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 34801 ENDING AT
36000,
  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 36001 ENDING AT
37200,
  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 37201 ENDING AT
38400,
  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_LINE33.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE33;
CREATE TABLE ORDER_LINE33

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_033
INDEX IN ts_orderline_033
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 38401 ENDING AT
39600,
  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_LINE34.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE34;
CREATE TABLE ORDER_LINE34

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_034
INDEX IN ts_orderline_034
ORGANIZE BY KEY SEQUENCE (
```

```
OL_W_ID STARTING FROM 39601 ENDING AT
40800,
  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_LINE35.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE35;
CREATE TABLE ORDER_LINE35

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_035
INDEX IN ts_orderline_035
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 40801 ENDING AT
42000,
  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_LINE36.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE36;
CREATE TABLE ORDER_LINE36

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
```

```
OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_036
INDEX IN ts_orderline_036
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 42001 ENDING AT
43200,
  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_LINE37.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE37;
CREATE TABLE ORDER_LINE37

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_037
INDEX IN ts_orderline_037
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 43201 ENDING AT
44400,
  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 LINE38.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE38;
CREATE TABLE ORDER_LINE38
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
IN ts_orderline_038
INDEX IN ts_orderline_038
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 44401 ENDING AT
45600,
  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 LINE39.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE39;
CREATE TABLE ORDER_LINE39
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
)
IN ts_orderline_039
INDEX IN ts_orderline_039
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 45601 ENDING AT
46800,
  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 2401 ENDING AT
3600,
  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 LINE40.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE40;
```

```
CREATE TABLE ORDER_LINE40
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
IN ts_orderline_040
INDEX IN ts_orderline_040
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 46801 ENDING AT
48000,
  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 LINE41.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE41;
CREATE TABLE ORDER_LINE41
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
IN ts_orderline_041
INDEX IN ts_orderline_041
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 48001 ENDING AT
49200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
```

```

3675, OL_O_ID STARTING FROM 1 ENDING AT
      OL_NUMBER STARTING FROM 1 ENDING AT
15    )
      ALLOW OVERFLOW;
connect reset;

```

CRTB ORDER LINE42.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE42;
CREATE TABLE ORDER_LINE42
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_042
INDEX IN ts_orderline_042
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 49201 ENDING AT
50400,
  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 LINE43.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE43;
CREATE TABLE ORDER_LINE43
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,

```

```

  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_043
INDEX IN ts_orderline_043
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 50401 ENDING AT
51600,
  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 LINE44.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE44;
CREATE TABLE ORDER_LINE44
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_044
INDEX IN ts_orderline_044
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 51601 ENDING AT
52800,
  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 LINE45.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE45;
CREATE TABLE ORDER_LINE45
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_045
INDEX IN ts_orderline_045
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 52801 ENDING AT
54000,
  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 LINE46.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE46;
CREATE TABLE ORDER_LINE46
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_046
INDEX IN ts_orderline_046
ORGANIZE BY KEY SEQUENCE (

```



```

55200, OL_W_ID STARTING FROM 54001 ENDING AT
      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_LINE47.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE47;
CREATE TABLE ORDER_LINE47
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_047
INDEX IN ts_orderline_047
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 55201 ENDING AT
55400,
  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_LINE48.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE48;
CREATE TABLE ORDER_LINE48
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,

```

```

      OL_I_ID INTEGER NOT NULL,
      OL_SUPPLY_W_ID INTEGER NOT NULL,
      OL_QUANTITY SMALLINT NOT NULL,
      OL_DIST_INFO CHAR(24) NOT NULL,
      OL_O_ID INTEGER NOT NULL,
      OL_D_ID SMALLINT NOT NULL,
      OL_W_ID INTEGER NOT NULL,
      OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_048
INDEX IN ts_orderline_048
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 56401 ENDING AT
57600,
  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_LINE49.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE49;
CREATE TABLE ORDER_LINE49
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_049
INDEX IN ts_orderline_049
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 57601 ENDING AT
58800,
  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 3601 ENDING AT
4800,
  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_LINE50.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE50;
CREATE TABLE ORDER_LINE50
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)

```

```

)
IN ts_orderline_050
INDEX IN ts_orderline_050
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 58801 ENDING AT
60000,
  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_LINE51.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE51;
CREATE TABLE ORDER_LINE51

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_051
INDEX IN ts_orderline_051
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 60001 ENDING AT
61200,
  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_LINE52.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE52;

```

```

CREATE TABLE ORDER_LINE52
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_052
INDEX IN ts_orderline_052
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 61201 ENDING AT
62400,
  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_LINE53.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE53;
CREATE TABLE ORDER_LINE53

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_053
INDEX IN ts_orderline_053
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 62401 ENDING AT
63600,
  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_LINE54.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE54;
CREATE TABLE ORDER_LINE54

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_054
INDEX IN ts_orderline_054
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 63601 ENDING AT
64800,
  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_LINE55.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE55;
CREATE TABLE ORDER_LINE55

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,

```

```

OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_055
INDEX IN ts_orderline_055
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 64801 ENDING AT
66000,
  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_LINE56.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE56;
CREATE TABLE ORDER_LINE56
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_056
INDEX IN ts_orderline_056
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 66001 ENDING AT
67200,
  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_LINE57.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE57;
CREATE TABLE ORDER_LINE57
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_057
INDEX IN ts_orderline_057
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 67201 ENDING AT
68400,
  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_LINE58.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE58;
CREATE TABLE ORDER_LINE58
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_058
INDEX IN ts_orderline_058
ORGANIZE BY KEY SEQUENCE (

```

```

  OL_W_ID STARTING FROM 68401 ENDING AT
69600,
  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_LINE59.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE59;
CREATE TABLE ORDER_LINE59
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_059
INDEX IN ts_orderline_059
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 69601 ENDING AT
70800,
  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 4801 ENDING AT
6000, 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 LINE60.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE60;
CREATE TABLE ORDER_LINE60
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_060
INDEX IN ts_orderline_060
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 70801 ENDING AT
72000, 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 LINE61.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE61;
CREATE TABLE ORDER_LINE61
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_061
INDEX IN ts_orderline_061
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 72001 ENDING AT
73200, 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 LINE62.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE62;
CREATE TABLE ORDER_LINE62
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)

```

```

)
IN ts_orderline_062
INDEX IN ts_orderline_062
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 73201 ENDING AT
74400, 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 LINE63.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE63;
CREATE TABLE ORDER_LINE63
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_063
INDEX IN ts_orderline_063
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 74401 ENDING AT
75600, 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 LINE64.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE64;

```

```

CREATE TABLE ORDER_LINE64
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_064
INDEX IN ts_orderline_064
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 75601 ENDING AT
76800,
  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 LINE65.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE65;
CREATE TABLE ORDER_LINE65
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_065
INDEX IN ts_orderline_065
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 76801 ENDING AT
78000,
  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 LINE66.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE66;
CREATE TABLE ORDER_LINE66
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_066
INDEX IN ts_orderline_066
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 78001 ENDING AT
79200,
  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 LINE67.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE67;
CREATE TABLE ORDER_LINE67
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,

```

```

  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_067
INDEX IN ts_orderline_067
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 79201 ENDING AT
80400,
  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 LINE68.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE68;
CREATE TABLE ORDER_LINE68
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_068
INDEX IN ts_orderline_068
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 80401 ENDING AT
81600,
  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_LINE69.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE69;
CREATE TABLE ORDER_LINE69

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

IN ts_orderline_069
INDEX IN ts_orderline_069
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 81601 ENDING AT

82800, 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 (

7200, OL_W_ID STARTING FROM 6001 ENDING AT

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_LINE70.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE70;
CREATE TABLE ORDER_LINE70

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

IN ts_orderline_070
INDEX IN ts_orderline_070
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 82801 ENDING AT

84000, 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_LINE71.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE71;
CREATE TABLE ORDER_LINE71

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
```

```
OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

IN ts_orderline_071
INDEX IN ts_orderline_071
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 84001 ENDING AT

85200, 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_LINE72.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE72;
CREATE TABLE ORDER_LINE72

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

IN ts_orderline_072
INDEX IN ts_orderline_072
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 85201 ENDING AT

86400, 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 LINE73.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE73;
CREATE TABLE ORDER_LINE73
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
IN ts_orderline_073
INDEX IN ts_orderline_073
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 86401 ENDING AT
87600,
  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 LINE74.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE74;
CREATE TABLE ORDER_LINE74
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
)
IN ts_orderline_074
INDEX IN ts_orderline_074
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 87601 ENDING AT
88800,
  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 LINE75.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE75;
CREATE TABLE ORDER_LINE75
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
IN ts_orderline_075
INDEX IN ts_orderline_075
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 88801 ENDING AT
90000,
  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 LINE76.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE76;
```

```
CREATE TABLE ORDER_LINE76
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
IN ts_orderline_076
INDEX IN ts_orderline_076
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 90001 ENDING AT
91200,
  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 LINE77.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE77;
CREATE TABLE ORDER_LINE77
```

```
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
```

```
IN ts_orderline_077
INDEX IN ts_orderline_077
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 91201 ENDING AT
92400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
```

```

3675, OL_O_ID STARTING FROM 1 ENDING AT
15 OL_NUMBER STARTING FROM 1 ENDING AT
)
ALLOW OVERFLOW;
connect reset;

```

CRTB ORDER LINE78.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE78;
CREATE TABLE ORDER_LINE78
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_078
INDEX IN ts_orderline_078
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 92401 ENDING AT
93600,
  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 LINE79.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE79;
CREATE TABLE ORDER_LINE79
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,

```

```

  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_079
INDEX IN ts_orderline_079
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 93601 ENDING AT
94800,
  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 7201 ENDING AT
8400,
  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 LINE80.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE80;
CREATE TABLE ORDER_LINE80
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_080
INDEX IN ts_orderline_080
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 94801 ENDING AT
96000,
  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 (

```



```

9600, OL_W_ID STARTING FROM 8401 ENDING AT
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 is_orderline_009
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 9601 ENDING AT
10800, 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 10801 ENDING AT
12000, 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 12001 ENDING AT
13200, 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 13201 ENDING AT
14400, 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 14401 ENDING AT
15600, 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 15601 ENDING AT
16800,
  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 16801 ENDING AT
18000,
  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 18001 ENDING AT
19200,
  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 19201 ENDING AT
20400,
  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 20401 ENDING AT
21600,
  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 21601 ENDING AT
22800,
  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 1200,
  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 22801 ENDING AT
24000,
  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 24001 ENDING AT
25200,
  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 25201 ENDING AT
26400,
  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 26401 ENDING AT
27600,
  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 27601 ENDING AT
28800,
  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 28801 ENDING AT
30000,
  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 30001 ENDING AT
31200,
  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 31201 ENDING AT
32400,
  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 32401 ENDING AT
33600,
  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 33601 ENDING AT
34800,
  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 1201 ENDING AT
2400,
  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 34801 ENDING AT
36000,
  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 36001 ENDING AT
37200,
  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 37201 ENDING AT
38400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS33.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS33;
CREATE TABLE ORDERS33
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_033
INDEX IN is_order_033
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 38401 ENDING AT
39600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS34.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS34;
CREATE TABLE ORDERS34
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_034
INDEX IN is_order_034
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 39601 ENDING AT
40800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS35.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS35;
CREATE TABLE ORDERS35
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_035
INDEX IN is_order_035
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 40801 ENDING AT
42000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS36.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS36;
CREATE TABLE ORDERS36
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_036
INDEX IN is_order_036
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 42001 ENDING AT
43200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS37.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS37;
CREATE TABLE ORDERS37
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_037
INDEX IN is_order_037
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 43201 ENDING AT
44400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS38.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS38;
CREATE TABLE ORDERS38
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_038
INDEX IN is_order_038
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 44401 ENDING AT
45600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS39.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS39;
CREATE TABLE ORDERS39
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_039
INDEX IN is_order_039
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 45601 ENDING AT
46800,
  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 2401 ENDING AT
3600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS40.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS40;
CREATE TABLE ORDERS40
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_040
INDEX IN is_order_040
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 46801 ENDING AT
48000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS41.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS41;
CREATE TABLE ORDERS41
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_041
INDEX IN is_order_041
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 48001 ENDING AT
49200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS42.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS42;
CREATE TABLE ORDERS42
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_042
INDEX IN is_order_042
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 49201 ENDING AT
50400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS43.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS43;
CREATE TABLE ORDERS43
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_043
INDEX IN is_order_043
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 50401 ENDING AT
51600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS44.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS44;
CREATE TABLE ORDERS44
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_044
INDEX IN is_order_044
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 51601 ENDING AT
52800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS45.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS45;
CREATE TABLE ORDERS45
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_045
INDEX IN is_order_045
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 52801 ENDING AT
54000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS46.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS46;
CREATE TABLE ORDERS46
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_046
INDEX IN is_order_046
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 54001 ENDING AT
55200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS47.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS47;
CREATE TABLE ORDERS47
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_047
INDEX IN is_order_047
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 55201 ENDING AT
56400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS48.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS48;
CREATE TABLE ORDERS48
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_048
INDEX IN is_order_048
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 56401 ENDING AT
57600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS49.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS49;
CREATE TABLE ORDERS49
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_049
INDEX IN is_order_049
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 57601 ENDING AT
58800,
  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 3601 ENDING AT
4800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS50.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS50;
CREATE TABLE ORDERS50
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_050
INDEX IN is_order_050
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 58801 ENDING AT
60000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS51.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS51;
CREATE TABLE ORDERS51
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_051
INDEX IN is_order_051
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 60001 ENDING AT
61200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS52.ddl


```

connect to TPCC in share mode;
DROP TABLE ORDERS52;
CREATE TABLE ORDERS52
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_052
INDEX IN is_order_052
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 61201 ENDING AT
62400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS53.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS53;
CREATE TABLE ORDERS53
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_053
INDEX IN is_order_053
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 62401 ENDING AT
63600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS54.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS54;
CREATE TABLE ORDERS54
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_054
INDEX IN is_order_054
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 63601 ENDING AT
64800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS55.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS55;
CREATE TABLE ORDERS55
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_055
INDEX IN is_order_055
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 64801 ENDING AT
66000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS56.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS56;
CREATE TABLE ORDERS56
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_056
INDEX IN is_order_056
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 66001 ENDING AT
67200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS57.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS57;
CREATE TABLE ORDERS57
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_057
INDEX IN is_order_057
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 67201 ENDING AT
68400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS58.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS58;
CREATE TABLE ORDERS58
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_058
INDEX IN is_order_058
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 68401 ENDING AT
69600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS59.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS59;
CREATE TABLE ORDERS59
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_059
INDEX IN is_order_059
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 69601 ENDING AT
70800,
  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 4801 ENDING AT
6000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS60.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS60;
CREATE TABLE ORDERS60
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_060
INDEX IN is_order_060
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 70801 ENDING AT
72000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS61.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS61;
CREATE TABLE ORDERS61
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_061
INDEX IN is_order_061
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 72001 ENDING AT
73200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS62.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS62;
CREATE TABLE ORDERS62
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_062
INDEX IN is_order_062
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 73201 ENDING AT
74400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS63.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS63;
CREATE TABLE ORDERS63
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_063
INDEX IN is_order_063
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 74401 ENDING AT
75600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS64.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS64;
CREATE TABLE ORDERS64
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_064
INDEX IN is_order_064
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 75601 ENDING AT
76800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS65.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS65;
CREATE TABLE ORDERS65
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_065
INDEX IN is_order_065
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 76801 ENDING AT
78000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS66.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS66;
CREATE TABLE ORDERS66
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_066
INDEX IN is_order_066
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 78001 ENDING AT
79200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS67.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS67;
CREATE TABLE ORDERS67
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_067
INDEX IN is_order_067
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 79201 ENDING AT
80400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS68.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS68;
CREATE TABLE ORDERS68
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_068
INDEX IN is_order_068
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 80401 ENDING AT
81600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS69.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS69;
CREATE TABLE ORDERS69
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_069
INDEX IN is_order_069
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 81601 ENDING AT
82800,
  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 6001 ENDING AT
7200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS70.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS70;
CREATE TABLE ORDERS70
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_070
INDEX IN is_order_070
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 82801 ENDING AT
84000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS71.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS71;
CREATE TABLE ORDERS71
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_071
INDEX IN is_order_071
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 84001 ENDING AT
85200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS72.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS72;
CREATE TABLE ORDERS72
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_072
INDEX IN is_order_072
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 85201 ENDING AT
86400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS73.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS73;
CREATE TABLE ORDERS73
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_073
INDEX IN is_order_073
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 86401 ENDING AT
87600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS74.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS74;
CREATE TABLE ORDERS74
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_074
INDEX IN is_order_074
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 87601 ENDING AT
88800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS75.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS75;
CREATE TABLE ORDERS75
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_075
INDEX IN is_order_075
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 88801 ENDING AT
90000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS76.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS76;
CREATE TABLE ORDERS76
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_076
INDEX IN is_order_076
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 90001 ENDING AT
91200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS77.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS77;
CREATE TABLE ORDERS77
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_077
INDEX IN is_order_077
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 91201 ENDING AT
92400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS78.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS78;
CREATE TABLE ORDERS78
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_078
INDEX IN is_order_078
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 92401 ENDING AT
93600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_ORDERS79.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS79;
CREATE TABLE ORDERS79
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_079
INDEX IN is_order_079
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 93601 ENDING AT
94800,
  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 7201 ENDING AT
```

8400,

```
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_ORDERS80.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDERS80;
CREATE TABLE ORDERS80
```

```
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D  TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID       INTEGER NOT NULL,
  O_W_ID     INTEGER NOT NULL,
  O_D_ID     SMALLINT NOT NULL
)
IN ts_order_080
INDEX IN is_order_080
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3675,
  O_W_ID STARTING FROM 94801 ENDING AT
```

96000,

```
  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 8401 ENDING AT
```

9600,

```
  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 9601 ENDING AT
```

10800,

```
  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 10801 ENDING AT
```

12000

```
)
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 12001 ENDING AT
13200
)
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 13201 ENDING AT
14400
)

```

```

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 14401 ENDING AT
15600
)
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 15601 ENDING AT
16800
)
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 16801 ENDING AT
18000
)
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 18001 ENDING AT
19200
)
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 19201 ENDING AT
20400
)
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 20401 ENDING AT
21600
)
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 21601 ENDING AT
22800
)
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 1200
)
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 22801 ENDING AT
24000
)
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 24001 ENDING AT
25200
)
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 25201 ENDING AT
26400
)
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 26401 ENDING AT
27600
)
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 27601 ENDING AT
28800
)
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 28801 ENDING AT
30000
)
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 30001 ENDING AT
31200
)
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 31201 ENDING AT
32400
)

```

```
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 32401 ENDING AT
33600
)
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 33601 ENDING AT
34800
)
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 1201 ENDING AT
2400
)
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 34801 ENDING AT
36000
)
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 36001 ENDING AT
37200
)
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 37201 ENDING AT
38400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK33.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK33;
CREATE TABLE STOCK33
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

)
IN ts_stock_033
INDEX IN ts_stock_033
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 38401 ENDING AT
39600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK34.ddl

```

connect to TPCC in share mode;

```

```

DROP TABLE STOCK34;
CREATE TABLE STOCK34
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_034
INDEX IN ts_stock_034
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 39601 ENDING AT
40800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK35.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK35;
CREATE TABLE STOCK35
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
)

```

```

S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_035
INDEX IN ts_stock_035
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 40801 ENDING AT
42000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK36.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK36;
CREATE TABLE STOCK36
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_036
INDEX IN ts_stock_036
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 42001 ENDING AT
43200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK37.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK37;
CREATE TABLE STOCK37
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_037
INDEX IN ts_stock_037
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 43201 ENDING AT
44400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK38.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK38;
CREATE TABLE STOCK38
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,

```

```

S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_038
INDEX IN ts_stock_038
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 44401 ENDING AT
45600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK39.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK39;
CREATE TABLE STOCK39
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_039
INDEX IN ts_stock_039
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 45601 ENDING AT
46800
)

```

```
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 2401 ENDING AT
3600
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK40.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK40;
CREATE TABLE STOCK40
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
```

```
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_040
INDEX IN ts_stock_040
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 46801 ENDING AT
48000
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK41.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK41;
CREATE TABLE STOCK41
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_041
INDEX IN ts_stock_041
ORGANIZE BY KEY SEQUENCE (
```

```
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 48001 ENDING AT
49200
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK42.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK42;
CREATE TABLE STOCK42
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_042
INDEX IN ts_stock_042
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 49201 ENDING AT
50400
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK43.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK43;
CREATE TABLE STOCK43
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
```

```

S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_043
INDEX IN ts_stock_043
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 50401 ENDING AT
51600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK44.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK44;
CREATE TABLE STOCK44
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

)
IN ts_stock_044
INDEX IN ts_stock_044
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 51601 ENDING AT
52800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK45.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK45;
CREATE TABLE STOCK45
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_045
INDEX IN ts_stock_045
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 52801 ENDING AT
54000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK46.ddl

```

connect to TPCC in share mode;

```

```

DROP TABLE STOCK46;
CREATE TABLE STOCK46
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_046
INDEX IN ts_stock_046
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 54001 ENDING AT
55200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK47.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK47;
CREATE TABLE STOCK47
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,

```

```

S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_047
INDEX IN ts_stock_047
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 55201 ENDING AT
56400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK48.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK48;
CREATE TABLE STOCK48
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_048
INDEX IN ts_stock_048
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 56401 ENDING AT
57600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK49.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK49;
CREATE TABLE STOCK49
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_049
INDEX IN ts_stock_049
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 57601 ENDING AT
58800
)
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 3601 ENDING AT
4800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK50.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK50;
CREATE TABLE STOCK50
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_050
INDEX IN ts_stock_050
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 58801 ENDING AT
60000
)

```



```
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK51.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK51;
CREATE TABLE STOCK51
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_051
INDEX IN ts_stock_051
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 60001 ENDING AT
61200
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK52.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK52;
CREATE TABLE STOCK52
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
```

```
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_052
INDEX IN ts_stock_052
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 61201 ENDING AT
62400
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK53.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK53;
CREATE TABLE STOCK53
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_053
INDEX IN ts_stock_053
ORGANIZE BY KEY SEQUENCE (
```

```
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 62401 ENDING AT
63600
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK54.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK54;
CREATE TABLE STOCK54
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_054
INDEX IN ts_stock_054
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 63601 ENDING AT
64800
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK55.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK55;
CREATE TABLE STOCK55
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
```

```

S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_055
INDEX IN ts_stock_055
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 64801 ENDING AT
66000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK56.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK56;
CREATE TABLE STOCK56
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

)
IN ts_stock_056
INDEX IN ts_stock_056
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 66001 ENDING AT
67200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK57.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK57;
CREATE TABLE STOCK57
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_057
INDEX IN ts_stock_057
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 67201 ENDING AT
68400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK58.ddl

```

connect to TPCC in share mode;

```

```

DROP TABLE STOCK58;
CREATE TABLE STOCK58
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_058
INDEX IN ts_stock_058
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 68401 ENDING AT
69600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK59.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK59;
CREATE TABLE STOCK59
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,

```

```

S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_059
INDEX IN ts_stock_059
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 69601 ENDING AT
70800
)
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 4801 ENDING AT
6000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK60.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK60;
CREATE TABLE STOCK60
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_060
INDEX IN ts_stock_060
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 70801 ENDING AT
72000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK61.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK61;
CREATE TABLE STOCK61
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,

```

```

S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_061
INDEX IN ts_stock_061
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 72001 ENDING AT
73200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK62.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK62;
CREATE TABLE STOCK62
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_062
INDEX IN ts_stock_062
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 73201 ENDING AT
74400
)

```

```
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK63.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK63;
CREATE TABLE STOCK63
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_063
INDEX IN ts_stock_063
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 74401 ENDING AT
75600
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK64.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK64;
CREATE TABLE STOCK64
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
```

```
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_064
INDEX IN ts_stock_064
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 75601 ENDING AT
76800
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK65.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK65;
CREATE TABLE STOCK65
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_065
INDEX IN ts_stock_065
ORGANIZE BY KEY SEQUENCE (
```

```
S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 76801 ENDING AT
78000
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK66.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK66;
CREATE TABLE STOCK66
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_066
INDEX IN ts_stock_066
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 78001 ENDING AT
79200
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK67.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK67;
CREATE TABLE STOCK67
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
```

```

S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_067
INDEX IN ts_stock_067
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 79201 ENDING AT
80400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK68.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK68;
CREATE TABLE STOCK68
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

)
IN ts_stock_068
INDEX IN ts_stock_068
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 80401 ENDING AT
81600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK69.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK69;
CREATE TABLE STOCK69
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_069
INDEX IN ts_stock_069
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 81601 ENDING AT
82800
)
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 6001 ENDING AT
7200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK70.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK70;
CREATE TABLE STOCK70
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,

```

```

S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_070
INDEX IN ts_stock_070
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 82801 ENDING AT
84000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK71.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK71;
CREATE TABLE STOCK71
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_071
INDEX IN ts_stock_071
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 84001 ENDING AT
85200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK72.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK72;
CREATE TABLE STOCK72
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_072
INDEX IN ts_stock_072
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 85201 ENDING AT
86400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK73.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK73;
CREATE TABLE STOCK73
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,

```

```

S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_073
INDEX IN ts_stock_073
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 86401 ENDING AT
87600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK74.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK74;
CREATE TABLE STOCK74
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_074
INDEX IN ts_stock_074
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 87601 ENDING AT
88800
)

```

```
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK75.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK75;
CREATE TABLE STOCK75
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_075
INDEX IN ts_stock_075
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 88801 ENDING AT
90000
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK76.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK76;
CREATE TABLE STOCK76
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
```

```
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_076
INDEX IN ts_stock_076
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 90001 ENDING AT
91200
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK77.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK77;
CREATE TABLE STOCK77
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_077
INDEX IN ts_stock_077
ORGANIZE BY KEY SEQUENCE (
```

```
S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 91201 ENDING AT
92400
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK78.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK78;
CREATE TABLE STOCK78
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)
```

```
IN ts_stock_078
INDEX IN ts_stock_078
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT
100000,
  S_W_ID STARTING FROM 92401 ENDING AT
93600
)
ALLOW OVERFLOW;
connect reset;
```

CRTB_STOCK79.ddl

```
connect to TPCC in share mode;
DROP TABLE STOCK79;
CREATE TABLE STOCK79
```

```
(
  S_REMOTE_CNT INTEGER NOT NULL,
```

```

S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_079
INDEX IN ts_stock_079
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 93601 ENDING AT
94800
)
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 7201 ENDING AT
8400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB_STOCK80.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK80;
CREATE TABLE STOCK80
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_080
INDEX IN ts_stock_080
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT
100000,
S_W_ID STARTING FROM 94801 ENDING AT
96000
)
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 8401 ENDING AT
9600
)
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 9601 ENDING AT
10800
)
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 10801 ENDING AT
12000
)
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 12001 ENDING AT
13200
)
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 13201 ENDING AT
14400
)
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 14401 ENDING AT
15600
)
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 15601 ENDING AT
16800
)
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 16801 ENDING AT
18000
)
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 18001 ENDING AT
19200
)
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 19201 ENDING AT
20400
)
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 20401 ENDING AT
21600
)
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 21601 ENDING AT
22800
)
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 1200
)
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 22801 ENDING AT
24000
)
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 24001 ENDING AT
25200
)
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 25201 ENDING AT
26400
)
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 26401 ENDING AT
27600
)
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 27601 ENDING AT
28800
)
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 28801 ENDING AT
30000
)
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 30001 ENDING AT
31200
)
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 31201 ENDING AT
32400
)
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 32401 ENDING AT
33600
)
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 33601 ENDING AT
34800
)
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 1201 ENDING AT
2400
)
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 34801 ENDING AT
36000
)
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 36001 ENDING AT
37200
)
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 37201 ENDING AT
38400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE33.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE33;
CREATE TABLE WAREHOUSE33

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_033
INDEX IN ts_wh_033
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 38401 ENDING AT
39600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE34.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE34;
CREATE TABLE WAREHOUSE34

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_034
INDEX IN ts_wh_034
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 39601 ENDING AT
40800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE35.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE35;
CREATE TABLE WAREHOUSE35

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_035
INDEX IN ts_wh_035
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 40801 ENDING AT
42000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE36.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE36;
CREATE TABLE WAREHOUSE36

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_036
INDEX IN ts_wh_036
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 42001 ENDING AT
43200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE37.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE37;
CREATE TABLE WAREHOUSE37

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_037
INDEX IN ts_wh_037
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 43201 ENDING AT
44400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE38.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE38;
CREATE TABLE WAREHOUSE38

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_038
INDEX IN ts_wh_038
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 44401 ENDING AT
45600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE39.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE39;
CREATE TABLE WAREHOUSE39

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_039
INDEX IN ts_wh_039
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 45601 ENDING AT
46800
)
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 2401 ENDING AT
3600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE40.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE40;
CREATE TABLE WAREHOUSE40

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_040
INDEX IN ts_wh_040
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 46801 ENDING AT
48000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE41.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE41;
CREATE TABLE WAREHOUSE41

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_041
INDEX IN ts_wh_041
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 48001 ENDING AT
49200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE42.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE42;
CREATE TABLE WAREHOUSE42

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_042
INDEX IN ts_wh_042
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 49201 ENDING AT
50400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE43.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE43;
CREATE TABLE WAREHOUSE43

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_043
INDEX IN ts_wh_043
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 50401 ENDING AT
51600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE44.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE44;
CREATE TABLE WAREHOUSE44

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_044
INDEX IN ts_wh_044
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 51601 ENDING AT
52800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE45.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE45;
CREATE TABLE WAREHOUSE45

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_045
INDEX IN ts_wh_045
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 52801 ENDING AT
54000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE46.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE46;
CREATE TABLE WAREHOUSE46

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_046
INDEX IN ts_wh_046
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 54001 ENDING AT
55200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE47.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE47;
CREATE TABLE WAREHOUSE47

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_047
INDEX IN ts_wh_047
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 55201 ENDING AT
56400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE48.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE48;
CREATE TABLE WAREHOUSE48

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_048
INDEX IN ts_wh_048
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 56401 ENDING AT
57600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE49.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE49;
CREATE TABLE WAREHOUSE49

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_049
INDEX IN ts_wh_049
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 57601 ENDING AT
58800
)
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 3601 ENDING AT
4800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE50.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE50;
CREATE TABLE WAREHOUSE50

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_050
INDEX IN ts_wh_050
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 58801 ENDING AT
60000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE51.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE51;
CREATE TABLE WAREHOUSE51

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_051
INDEX IN ts_wh_051
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 60001 ENDING AT
61200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE52.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE52;
CREATE TABLE WAREHOUSE52

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_052
INDEX IN ts_wh_052
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 61201 ENDING AT
62400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE53.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE53;
CREATE TABLE WAREHOUSE53

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```



```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_053
INDEX IN ts_wh_053
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 62401 ENDING AT
63600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE54.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE54;
CREATE TABLE WAREHOUSE54

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_054
INDEX IN ts_wh_054
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 63601 ENDING AT
64800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE55.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE55;
CREATE TABLE WAREHOUSE55

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_055
INDEX IN ts_wh_055
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 64801 ENDING AT
66000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE56.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE56;
CREATE TABLE WAREHOUSE56

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_056
INDEX IN ts_wh_056
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 66001 ENDING AT
67200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE57.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE57;
CREATE TABLE WAREHOUSE57

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_057
INDEX IN ts_wh_057
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 67201 ENDING AT
68400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE58.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE58;
CREATE TABLE WAREHOUSE58

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_058
INDEX IN ts_wh_058
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 68401 ENDING AT
69600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE59.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE59;
CREATE TABLE WAREHOUSE59

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_059
INDEX IN ts_wh_059
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 69601 ENDING AT
70800
)
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 4801 ENDING AT
6000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE60.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE60;
CREATE TABLE WAREHOUSE60

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_060
INDEX IN ts_wh_060
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 70801 ENDING AT
72000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE61.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE61;
CREATE TABLE WAREHOUSE61

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_061
INDEX IN ts_wh_061
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 72001 ENDING AT
73200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE62.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE62;
CREATE TABLE WAREHOUSE62

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_062
INDEX IN ts_wh_062
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 73201 ENDING AT
74400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE63.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE63;
CREATE TABLE WAREHOUSE63

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_063
INDEX IN ts_wh_063
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 74401 ENDING AT
75600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE64.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE64;
CREATE TABLE WAREHOUSE64

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_064
INDEX IN ts_wh_064
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 75601 ENDING AT
76800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE65.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE65;
CREATE TABLE WAREHOUSE65

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_065
INDEX IN ts_wh_065
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 76801 ENDING AT
78000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE66.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE66;
CREATE TABLE WAREHOUSE66

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_066
INDEX IN ts_wh_066
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 78001 ENDING AT
79200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE67.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE67;
CREATE TABLE WAREHOUSE67

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_067
INDEX IN ts_wh_067
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 79201 ENDING AT
80400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE68.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE68;
CREATE TABLE WAREHOUSE68

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_068
INDEX IN ts_wh_068
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 80401 ENDING AT
81600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE69.ddl

connect to TPCC in share mode;
DROP TABLE WAREHOUSE69;
CREATE TABLE WAREHOUSE69

```

(W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_069
INDEX IN ts_wh_069
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 81601 ENDING AT
82800
)
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 6001 ENDING AT
7200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE70.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE70;
CREATE TABLE WAREHOUSE70

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_070
INDEX IN ts_wh_070
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 82801 ENDING AT
84000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE71.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE71;
CREATE TABLE WAREHOUSE71

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_071
INDEX IN ts_wh_071
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 84001 ENDING AT
85200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE72.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE72;
CREATE TABLE WAREHOUSE72

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_072
INDEX IN ts_wh_072
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 85201 ENDING AT
86400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE73.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE73;
CREATE TABLE WAREHOUSE73

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_073
INDEX IN ts_wh_073
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 86401 ENDING AT
87600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE74.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE74;
CREATE TABLE WAREHOUSE74

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_074
INDEX IN ts_wh_074
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 87601 ENDING AT
88800
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE75.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE75;
CREATE TABLE WAREHOUSE75

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_075
INDEX IN ts_wh_075
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 88801 ENDING AT
90000
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE76.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE76;
CREATE TABLE WAREHOUSE76

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_076
INDEX IN ts_wh_076
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 90001 ENDING AT
91200
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE77.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE77;
CREATE TABLE WAREHOUSE77

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_077
INDEX IN ts_wh_077
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 91201 ENDING AT
92400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE78.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE78;
CREATE TABLE WAREHOUSE78

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_078
INDEX IN ts_wh_078
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 92401 ENDING AT
93600
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE79.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE79;
CREATE TABLE WAREHOUSE79

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_079
INDEX IN ts_wh_079
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 93601 ENDING AT
94800
)
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 7201 ENDING AT
8400
)
ALLOW OVERFLOW;
connect reset;

```

CRTB WAREHOUSE80.ddl

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE80;
CREATE TABLE WAREHOUSE80

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_wh_080
INDEX IN ts_wh_080
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 94801 ENDING AT
96000
)
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 8401 ENDING AT
9600
)
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 9601 ENDING AT
10800
)
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 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38 UNION ALL
SELECT * FROM CUSTOMER39 UNION ALL
SELECT * FROM CUSTOMER40 UNION ALL
SELECT * FROM CUSTOMER41 UNION ALL
SELECT * FROM CUSTOMER42 UNION ALL
SELECT * FROM CUSTOMER43 UNION ALL
SELECT * FROM CUSTOMER44 UNION ALL
SELECT * FROM CUSTOMER45 UNION ALL
SELECT * FROM CUSTOMER46 UNION ALL
SELECT * FROM CUSTOMER47 UNION ALL
SELECT * FROM CUSTOMER48 UNION ALL
SELECT * FROM CUSTOMER49 UNION ALL
SELECT * FROM CUSTOMER50 UNION ALL
SELECT * FROM CUSTOMER51 UNION ALL
SELECT * FROM CUSTOMER52 UNION ALL
SELECT * FROM CUSTOMER53 UNION ALL
SELECT * FROM CUSTOMER54 UNION ALL
SELECT * FROM CUSTOMER55 UNION ALL
SELECT * FROM CUSTOMER56 UNION ALL
SELECT * FROM CUSTOMER57 UNION ALL
SELECT * FROM CUSTOMER58 UNION ALL
SELECT * FROM CUSTOMER59 UNION ALL
SELECT * FROM CUSTOMER60 UNION ALL
SELECT * FROM CUSTOMER61 UNION ALL

```

```

SELECT * FROM CUSTOMER62 UNION ALL
SELECT * FROM CUSTOMER63 UNION ALL
SELECT * FROM CUSTOMER64 UNION ALL
SELECT * FROM CUSTOMER65 UNION ALL
SELECT * FROM CUSTOMER66 UNION ALL
SELECT * FROM CUSTOMER67 UNION ALL
SELECT * FROM CUSTOMER68 UNION ALL
SELECT * FROM CUSTOMER69 UNION ALL
SELECT * FROM CUSTOMER70 UNION ALL
SELECT * FROM CUSTOMER71 UNION ALL
SELECT * FROM CUSTOMER72 UNION ALL
SELECT * FROM CUSTOMER73 UNION ALL
SELECT * FROM CUSTOMER74 UNION ALL
SELECT * FROM CUSTOMER75 UNION ALL
SELECT * FROM CUSTOMER76 UNION ALL
SELECT * FROM CUSTOMER77 UNION ALL
SELECT * FROM CUSTOMER78 UNION ALL
SELECT * FROM CUSTOMER79 UNION ALL
SELECT * FROM CUSTOMER80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW_DISTRICT.ddl

```

connect to TPCC in share mode;
DROP VIEW DISTRICT;
CREATE VIEW DISTRICT
(D_NEXT_O_ID,
D_TAX,
D_YTD,
D_NAME,
D_STREET_1,
D_STREET_2,
D_CITY,
D_STATE,
D_ZIP,
D_ID,
D_W_ID
) AS SELECT * FROM DISTRICT1 UNION ALL
SELECT * FROM DISTRICT2 UNION ALL
SELECT * FROM DISTRICT3 UNION ALL
SELECT * FROM DISTRICT4 UNION ALL
SELECT * FROM DISTRICT5 UNION ALL
SELECT * FROM DISTRICT6 UNION ALL
SELECT * FROM DISTRICT7 UNION ALL
SELECT * FROM DISTRICT8 UNION ALL
SELECT * FROM DISTRICT9 UNION ALL
SELECT * FROM DISTRICT10 UNION ALL
SELECT * FROM DISTRICT11 UNION ALL
SELECT * FROM DISTRICT12 UNION ALL

```

```

SELECT * FROM DISTRICT13 UNION ALL
SELECT * FROM DISTRICT14 UNION ALL
SELECT * FROM DISTRICT15 UNION ALL
SELECT * FROM DISTRICT16 UNION ALL
SELECT * FROM DISTRICT17 UNION ALL
SELECT * FROM DISTRICT18 UNION ALL
SELECT * FROM DISTRICT19 UNION ALL
SELECT * FROM DISTRICT20 UNION ALL
SELECT * FROM DISTRICT21 UNION ALL
SELECT * FROM DISTRICT22 UNION ALL
SELECT * FROM DISTRICT23 UNION ALL
SELECT * FROM DISTRICT24 UNION ALL
SELECT * FROM DISTRICT25 UNION ALL
SELECT * FROM DISTRICT26 UNION ALL
SELECT * FROM DISTRICT27 UNION ALL
SELECT * FROM DISTRICT28 UNION ALL
SELECT * FROM DISTRICT29 UNION ALL
SELECT * FROM DISTRICT30 UNION ALL
SELECT * FROM DISTRICT31 UNION ALL
SELECT * FROM DISTRICT32 UNION ALL
SELECT * FROM DISTRICT33 UNION ALL
SELECT * FROM DISTRICT34 UNION ALL
SELECT * FROM DISTRICT35 UNION ALL
SELECT * FROM DISTRICT36 UNION ALL
SELECT * FROM DISTRICT37 UNION ALL
SELECT * FROM DISTRICT38 UNION ALL
SELECT * FROM DISTRICT39 UNION ALL
SELECT * FROM DISTRICT40 UNION ALL
SELECT * FROM DISTRICT41 UNION ALL
SELECT * FROM DISTRICT42 UNION ALL
SELECT * FROM DISTRICT43 UNION ALL
SELECT * FROM DISTRICT44 UNION ALL
SELECT * FROM DISTRICT45 UNION ALL
SELECT * FROM DISTRICT46 UNION ALL
SELECT * FROM DISTRICT47 UNION ALL
SELECT * FROM DISTRICT48 UNION ALL
SELECT * FROM DISTRICT49 UNION ALL
SELECT * FROM DISTRICT50 UNION ALL
SELECT * FROM DISTRICT51 UNION ALL
SELECT * FROM DISTRICT52 UNION ALL
SELECT * FROM DISTRICT53 UNION ALL
SELECT * FROM DISTRICT54 UNION ALL
SELECT * FROM DISTRICT55 UNION ALL
SELECT * FROM DISTRICT56 UNION ALL
SELECT * FROM DISTRICT57 UNION ALL
SELECT * FROM DISTRICT58 UNION ALL
SELECT * FROM DISTRICT59 UNION ALL
SELECT * FROM DISTRICT60 UNION ALL
SELECT * FROM DISTRICT61 UNION ALL
SELECT * FROM DISTRICT62 UNION ALL
SELECT * FROM DISTRICT63 UNION ALL
SELECT * FROM DISTRICT64 UNION ALL

```

```

SELECT * FROM DISTRICT65 UNION ALL
SELECT * FROM DISTRICT66 UNION ALL
SELECT * FROM DISTRICT67 UNION ALL
SELECT * FROM DISTRICT68 UNION ALL
SELECT * FROM DISTRICT69 UNION ALL
SELECT * FROM DISTRICT70 UNION ALL
SELECT * FROM DISTRICT71 UNION ALL
SELECT * FROM DISTRICT72 UNION ALL
SELECT * FROM DISTRICT73 UNION ALL
SELECT * FROM DISTRICT74 UNION ALL
SELECT * FROM DISTRICT75 UNION ALL
SELECT * FROM DISTRICT76 UNION ALL
SELECT * FROM DISTRICT77 UNION ALL
SELECT * FROM DISTRICT78 UNION ALL
SELECT * FROM DISTRICT79 UNION ALL
SELECT * FROM DISTRICT80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

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 ORDER_LINE56 UNION ALL
SELECT * FROM ORDER_LINE57 UNION ALL
SELECT * FROM ORDER_LINE58 UNION ALL
SELECT * FROM ORDER_LINE59 UNION ALL
SELECT * FROM ORDER_LINE60 UNION ALL
SELECT * FROM ORDER_LINE61 UNION ALL
SELECT * FROM ORDER_LINE62 UNION ALL
SELECT * FROM ORDER_LINE63 UNION ALL
SELECT * FROM ORDER_LINE64 UNION ALL
SELECT * FROM ORDER_LINE65 UNION ALL
SELECT * FROM ORDER_LINE66 UNION ALL
SELECT * FROM ORDER_LINE67 UNION ALL
SELECT * FROM ORDER_LINE68 UNION ALL
SELECT * FROM ORDER_LINE69 UNION ALL
SELECT * FROM ORDER_LINE70 UNION ALL
SELECT * FROM ORDER_LINE71 UNION ALL
SELECT * FROM ORDER_LINE72 UNION ALL
SELECT * FROM ORDER_LINE73 UNION ALL
SELECT * FROM ORDER_LINE74 UNION ALL
SELECT * FROM ORDER_LINE75 UNION ALL
SELECT * FROM ORDER_LINE76 UNION ALL
SELECT * FROM ORDER_LINE77 UNION ALL
SELECT * FROM ORDER_LINE78 UNION ALL
SELECT * FROM ORDER_LINE79 UNION ALL
SELECT * FROM ORDER_LINE80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW ORDERS.ddl

```

connect to TPCC in share mode;
DROP VIEW ORDERS;
CREATE VIEW ORDERS
(O_C_ID,
 O_ENTRY_D,
 O_CARRIER_ID,
 O_OL_CNT,
 O_ALL_LOCAL,
 O_ID,
 O_W_ID,
 O_D_ID
) AS SELECT * FROM ORDERS1 UNION ALL
SELECT * FROM ORDERS2 UNION ALL
SELECT * FROM ORDERS3 UNION ALL
SELECT * FROM ORDERS4 UNION ALL
SELECT * FROM ORDERS5 UNION ALL
SELECT * FROM ORDERS6 UNION ALL
SELECT * FROM ORDERS7 UNION ALL
SELECT * FROM ORDERS8 UNION ALL
SELECT * FROM ORDERS9 UNION ALL

```

```

SELECT * FROM ORDERS10 UNION ALL
SELECT * FROM ORDERS11 UNION ALL
SELECT * FROM ORDERS12 UNION ALL
SELECT * FROM ORDERS13 UNION ALL
SELECT * FROM ORDERS14 UNION ALL
SELECT * FROM ORDERS15 UNION ALL
SELECT * FROM ORDERS16 UNION ALL
SELECT * FROM ORDERS17 UNION ALL
SELECT * FROM ORDERS18 UNION ALL
SELECT * FROM ORDERS19 UNION ALL
SELECT * FROM ORDERS20 UNION ALL
SELECT * FROM ORDERS21 UNION ALL
SELECT * FROM ORDERS22 UNION ALL
SELECT * FROM ORDERS23 UNION ALL
SELECT * FROM ORDERS24 UNION ALL
SELECT * FROM ORDERS25 UNION ALL
SELECT * FROM ORDERS26 UNION ALL
SELECT * FROM ORDERS27 UNION ALL
SELECT * FROM ORDERS28 UNION ALL
SELECT * FROM ORDERS29 UNION ALL
SELECT * FROM ORDERS30 UNION ALL
SELECT * FROM ORDERS31 UNION ALL
SELECT * FROM ORDERS32 UNION ALL
SELECT * FROM ORDERS33 UNION ALL
SELECT * FROM ORDERS34 UNION ALL
SELECT * FROM ORDERS35 UNION ALL
SELECT * FROM ORDERS36 UNION ALL
SELECT * FROM ORDERS37 UNION ALL
SELECT * FROM ORDERS38 UNION ALL
SELECT * FROM ORDERS39 UNION ALL
SELECT * FROM ORDERS40 UNION ALL
SELECT * FROM ORDERS41 UNION ALL
SELECT * FROM ORDERS42 UNION ALL
SELECT * FROM ORDERS43 UNION ALL
SELECT * FROM ORDERS44 UNION ALL
SELECT * FROM ORDERS45 UNION ALL
SELECT * FROM ORDERS46 UNION ALL
SELECT * FROM ORDERS47 UNION ALL
SELECT * FROM ORDERS48 UNION ALL
SELECT * FROM ORDERS49 UNION ALL
SELECT * FROM ORDERS50 UNION ALL
SELECT * FROM ORDERS51 UNION ALL
SELECT * FROM ORDERS52 UNION ALL
SELECT * FROM ORDERS53 UNION ALL
SELECT * FROM ORDERS54 UNION ALL
SELECT * FROM ORDERS55 UNION ALL
SELECT * FROM ORDERS56 UNION ALL
SELECT * FROM ORDERS57 UNION ALL
SELECT * FROM ORDERS58 UNION ALL
SELECT * FROM ORDERS59 UNION ALL
SELECT * FROM ORDERS60 UNION ALL
SELECT * FROM ORDERS61 UNION ALL

```

```

SELECT * FROM ORDERS62 UNION ALL
SELECT * FROM ORDERS63 UNION ALL
SELECT * FROM ORDERS64 UNION ALL
SELECT * FROM ORDERS65 UNION ALL
SELECT * FROM ORDERS66 UNION ALL
SELECT * FROM ORDERS67 UNION ALL
SELECT * FROM ORDERS68 UNION ALL
SELECT * FROM ORDERS69 UNION ALL
SELECT * FROM ORDERS70 UNION ALL
SELECT * FROM ORDERS71 UNION ALL
SELECT * FROM ORDERS72 UNION ALL
SELECT * FROM ORDERS73 UNION ALL
SELECT * FROM ORDERS74 UNION ALL
SELECT * FROM ORDERS75 UNION ALL
SELECT * FROM ORDERS76 UNION ALL
SELECT * FROM ORDERS77 UNION ALL
SELECT * FROM ORDERS78 UNION ALL
SELECT * FROM ORDERS79 UNION ALL
SELECT * FROM ORDERS80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

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 WAREHOUSE63 UNION ALL
SELECT * FROM WAREHOUSE64 UNION ALL
SELECT * FROM WAREHOUSE65 UNION ALL
SELECT * FROM WAREHOUSE66 UNION ALL
SELECT * FROM WAREHOUSE67 UNION ALL
SELECT * FROM WAREHOUSE68 UNION ALL
SELECT * FROM WAREHOUSE69 UNION ALL
SELECT * FROM WAREHOUSE70 UNION ALL
SELECT * FROM WAREHOUSE71 UNION ALL
SELECT * FROM WAREHOUSE72 UNION ALL
SELECT * FROM WAREHOUSE73 UNION ALL
SELECT * FROM WAREHOUSE74 UNION ALL
SELECT * FROM WAREHOUSE75 UNION ALL
SELECT * FROM WAREHOUSE76 UNION ALL
SELECT * FROM WAREHOUSE77 UNION ALL
SELECT * FROM WAREHOUSE78 UNION ALL
SELECT * FROM WAREHOUSE79 UNION ALL
SELECT * FROM WAREHOUSE80
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

GEN CUSTOMER 10.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10801 12000 -f1
/flx8/flat_010/customer_010_1.dat
```

GEN CUSTOMER 11.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12001 13200 -f1
/flx8/flat_011/customer_011_1.dat
```

GEN CUSTOMER 12.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13201 14400 -f1
/flx8/flat_012/customer_012_1.dat
```

GEN CUSTOMER 13.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14401 15600 -f1
/flx8/flat_013/customer_013_1.dat
```

GEN CUSTOMER 14.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15601 16800 -f1
/flx8/flat_014/customer_014_1.dat
```

GEN CUSTOMER 15.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16801 18000 -f1
/flx8/flat_015/customer_015_1.dat
```

GEN CUSTOMER 16.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18001 19200 -f1
/flx8/flat_016/customer_016_1.dat
```

GEN CUSTOMER 17.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19201 20400 -f1
/flx8/flat_017/customer_017_1.dat
```

GEN CUSTOMER 18.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20401 21600 -f1
/flx8/flat_018/customer_018_1.dat
```

GEN CUSTOMER 19.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21601 22800 -f1
/flx8/flat_019/customer_019_1.dat
```

GEN CUSTOMER 1.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 1200 -f1
/flx8/flat_001/customer_001_1.dat
```

GEN CUSTOMER 20.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22801 24000 -f1
/flx8/flat_020/customer_020_1.dat
```

GEN CUSTOMER 21.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24001 25200 -f1
/flx8/flat_021/customer_021_1.dat
```

GEN CUSTOMER 22.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25201 26400 -f1
/flx8/flat_022/customer_022_1.dat
```

GEN CUSTOMER 23.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26401 27600 -f1
/flx8/flat_023/customer_023_1.dat
```

GEN CUSTOMER 24.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27601 28800 -f1
/flx8/flat_024/customer_024_1.dat
```

GEN CUSTOMER 25.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28801 30000 -f1
/flx8/flat_025/customer_025_1.dat
```

GEN CUSTOMER 26.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30001 31200 -f1
/flx8/flat_026/customer_026_1.dat
```

GEN CUSTOMER 27.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31201 32400 -f1
/flx8/flat_027/customer_027_1.dat
```

GEN CUSTOMER 28.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32401 33600 -f1
/flx8/flat_028/customer_028_1.dat
```

GEN CUSTOMER 29.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33601 34800 -f1
/flx8/flat_029/customer_029_1.dat
```

GEN CUSTOMER 2.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1201 2400 -f1
/flx8/flat_002/customer_002_1.dat
```

GEN CUSTOMER 30.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34801 36000 -f1
/flx8/flat_030/customer_030_1.dat

GEN CUSTOMER 31.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36001 37200 -f1
/flx8/flat_031/customer_031_1.dat

GEN CUSTOMER 32.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37201 38400 -f1
/flx8/flat_032/customer_032_1.dat

GEN CUSTOMER 33.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38401 39600 -f1
/flx8/flat_033/customer_033_1.dat

GEN CUSTOMER 34.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39601 40800 -f1
/flx8/flat_034/customer_034_1.dat

GEN CUSTOMER 35.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40801 42000 -f1
/flx8/flat_035/customer_035_1.dat

GEN CUSTOMER 36.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 42001 43200 -f1
/flx8/flat_036/customer_036_1.dat

GEN CUSTOMER 37.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43201 44400 -f1
/flx8/flat_037/customer_037_1.dat

GEN CUSTOMER 38.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 44401 45600 -f1
/flx8/flat_038/customer_038_1.dat

GEN CUSTOMER 39.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 45601 46800 -f1
/flx8/flat_039/customer_039_1.dat

GEN CUSTOMER 3.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2401 3600 -f1
/flx8/flat_003/customer_003_1.dat

GEN CUSTOMER 40.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 46801 48000 -f1
/flx8/flat_040/customer_040_1.dat

GEN CUSTOMER 41.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 48001 49200 -f1
/flx8/flat_041/customer_041_1.dat

GEN CUSTOMER 42.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 49201 50400 -f1
/flx8/flat_042/customer_042_1.dat

GEN CUSTOMER 43.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50401 51600 -f1
/flx8/flat_043/customer_043_1.dat

GEN CUSTOMER 44.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 51601 52800 -f1
/flx8/flat_044/customer_044_1.dat

GEN CUSTOMER 45.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 52801 54000 -f1
/flx8/flat_045/customer_045_1.dat

GEN CUSTOMER 46.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 54001 55200 -f1
/flx8/flat_046/customer_046_1.dat

GEN CUSTOMER 47.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 55201 56400 -f1
/flx8/flat_047/customer_047_1.dat

GEN CUSTOMER 48.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 56401 57600 -f1
/flx8/flat_048/customer_048_1.dat

GEN CUSTOMER 49.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 57601 58800 -f1
/flx8/flat_049/customer_049_1.dat

GEN CUSTOMER 4.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3601 4800 -f1
/flx8/flat_004/customer_004_1.dat

GEN CUSTOMER 50.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 58801 60000 -f1
/flx8/flat_050/customer_050_1.dat

GEN CUSTOMER 51.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 60001 61200 -f1
/flx8/flat_051/customer_051_1.dat

GEN CUSTOMER 52.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 61201 62400 -f1
/flx8/flat_052/customer_052_1.dat

GEN CUSTOMER 53.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 62401 63600 -f1
/flx8/flat_053/customer_053_1.dat

GEN CUSTOMER 54.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 63601 64800 -f1
/flx8/flat_054/customer_054_1.dat

GEN CUSTOMER 55.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 64801 66000 -f1
/flx8/flat_055/customer_055_1.dat

GEN CUSTOMER 56.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 66001 67200 -f1
/flx8/flat_056/customer_056_1.dat

GEN CUSTOMER 57.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 67201 68400 -f1
/flx8/flat_057/customer_057_1.dat

GEN CUSTOMER 58.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 68401 69600 -f1
/flx8/flat_058/customer_058_1.dat

GEN CUSTOMER 59.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 69601 70800 -f1
/flx8/flat_059/customer_059_1.dat

GEN CUSTOMER 5.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4801 6000 -f1
/flx8/flat_005/customer_005_1.dat

GEN CUSTOMER 60.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 70801 72000 -f1
/flx8/flat_060/customer_060_1.dat

GEN CUSTOMER 61.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 72001 73200 -f1
/flx8/flat_061/customer_061_1.dat

GEN CUSTOMER 62.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 73201 74400 -f1
/flx8/flat_062/customer_062_1.dat

GEN CUSTOMER 63.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 74401 75600 -f1
/flx8/flat_063/customer_063_1.dat

GEN CUSTOMER 64.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 75601 76800 -f1
/flx8/flat_064/customer_064_1.dat

GEN CUSTOMER 65.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 76801 78000 -f1
/flx8/flat_065/customer_065_1.dat

GEN CUSTOMER 66.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 78001 79200 -f1
/flx8/flat_066/customer_066_1.dat

GEN CUSTOMER 67.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 79201 80400 -f1
/flx8/flat_067/customer_067_1.dat

GEN CUSTOMER 68.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 80401 81600 -f1
/flx8/flat_068/customer_068_1.dat

GEN CUSTOMER 69.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 81601 82800 -f1
/flx8/flat_069/customer_069_1.dat

GEN CUSTOMER 6.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6001 7200 -f1
/flx8/flat_006/customer_006_1.dat

GEN CUSTOMER 70.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 82801 84000 -f1
/flx8/flat_070/customer_070_1.dat

GEN CUSTOMER 71.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 84001 85200 -f1
/flx8/flat_071/customer_071_1.dat

GEN CUSTOMER 72.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 85201 86400 -f1
/flx8/flat_072/customer_072_1.dat

GEN CUSTOMER 73.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 86401 87600 -f1
/flx8/flat_073/customer_073_1.dat

GEN CUSTOMER 74.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 87601 88800 -f1
/flx8/flat_074/customer_074_1.dat

GEN CUSTOMER 75.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 88801 90000 -f1
/flx8/flat_075/customer_075_1.dat

GEN CUSTOMER 76.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 90001 91200 -f1
/flx8/flat_076/customer_076_1.dat

GEN CUSTOMER 77.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 91201 92400 -f1
/flx8/flat_077/customer_077_1.dat

GEN CUSTOMER 78.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 92401 93600 -f1
/flx8/flat_078/customer_078_1.dat

GEN CUSTOMER 79.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 93601 94800 -f1
/flx8/flat_079/customer_079_1.dat

GEN CUSTOMER 7.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7201 8400 -f1
/flx8/flat_007/customer_007_1.dat

GEN CUSTOMER 80.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 94801 96000 -f1
/flx8/flat_080/customer_080_1.dat

GEN CUSTOMER 8.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8401 9600 -f1
/flx8/flat_008/customer_008_1.dat

GEN CUSTOMER 9.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9601 10800 -f1
/flx8/flat_009/customer_009_1.dat

GEN DISTRICT 10.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 10801 12000 -f1
/flx8/flat_010/district_010_1.dat

GEN DISTRICT 11.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 12001 13200 -f1
/flx8/flat_011/district_011_1.dat

GEN DISTRICT 12.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 13201 14400 -f1
/flx8/flat_012/district_012_1.dat

GEN DISTRICT 13.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 14401 15600 -f1
/flx8/flat_013/district_013_1.dat

GEN DISTRICT 14.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 15601 16800 -f1
/flx8/flat_014/district_014_1.dat

GEN DISTRICT 15.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 16801 18000 -f1
/flx8/flat_015/district_015_1.dat

GEN DISTRICT 16.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 18001 19200 -f1
/flx8/flat_016/district_016_1.dat

GEN DISTRICT 17.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 19201 20400 -f1
/flx8/flat_017/district_017_1.dat

GEN DISTRICT 18.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 20401 21600 -f1
/flx8/flat_018/district_018_1.dat

GEN DISTRICT 19.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 21601 22800 -f1
/flx8/flat_019/district_019_1.dat

GEN DISTRICT 1.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1 1200 -f1
/flx8/flat_001/district_001_1.dat

GEN DISTRICT 20.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 22801 24000 -f1
/flx8/flat_020/district_020_1.dat

GEN DISTRICT 21.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 24001 25200 -f1
/flx8/flat_021/district_021_1.dat

GEN DISTRICT 22.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 25201 26400 -f1
/flx8/flat_022/district_022_1.dat

GEN DISTRICT 23.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 26401 27600 -f1
/flx8/flat_023/district_023_1.dat

GEN DISTRICT 24.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 27601 28800 -f1
/flx8/flat_024/district_024_1.dat

GEN DISTRICT 25.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 28801 30000 -f1
/flx8/flat_025/district_025_1.dat

GEN DISTRICT 26.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 30001 31200 -f1
/flx8/flat_026/district_026_1.dat

GEN DISTRICT 27.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 31201 32400 -f1
/flx8/flat_027/district_027_1.dat

GEN DISTRICT 28.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 32401 33600 -f1
/flx8/flat_028/district_028_1.dat

GEN DISTRICT 29.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 33601 34800 -f1
/flx8/flat_029/district_029_1.dat

GEN DISTRICT 2.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1201 2400 -f1
/flx8/flat_002/district_002_1.dat

GEN DISTRICT 30.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 34801 36000 -f1
/flx8/flat_030/district_030_1.dat

GEN DISTRICT 31.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 36001 37200 -f1
/flx8/flat_031/district_031_1.dat

GEN DISTRICT 32.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 37201 38400 -f1
/flx8/flat_032/district_032_1.dat

GEN DISTRICT 33.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 38401 39600 -f1
/flx8/flat_033/district_033_1.dat

GEN DISTRICT 34.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 39601 40800 -f1
/flx8/flat_034/district_034_1.dat

GEN DISTRICT 35.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 40801 42000 -f1
/flx8/flat_035/district_035_1.dat

GEN DISTRICT 36.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 42001 43200 -f1
/flx8/flat_036/district_036_1.dat

GEN DISTRICT 37.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 43201 44400 -f1
/flx8/flat_037/district_037_1.dat

GEN DISTRICT 38.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 44401 45600 -f1
/flx8/flat_038/district_038_1.dat

GEN DISTRICT 39.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 45601 46800 -f1
/flx8/flat_039/district_039_1.dat

GEN DISTRICT 3.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 2401 3600 -f1
/flx8/flat_003/district_003_1.dat

GEN DISTRICT 40.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 46801 48000 -f1
/flx8/flat_040/district_040_1.dat

GEN DISTRICT 41.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 48001 49200 -f1
/flx8/flat_041/district_041_1.dat

GEN DISTRICT 42.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 49201 50400 -f1
/flx8/flat_042/district_042_1.dat

GEN DISTRICT 43.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 50401 51600 -f1
/flx8/flat_043/district_043_1.dat

GEN DISTRICT 44.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 51601 52800 -f1
/flx8/flat_044/district_044_1.dat

GEN DISTRICT 45.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 52801 54000 -f1
/flx8/flat_045/district_045_1.dat

GEN DISTRICT 46.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 54001 55200 -f1
/flx8/flat_046/district_046_1.dat

GEN DISTRICT 47.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 55201 56400 -f1
/flx8/flat_047/district_047_1.dat

GEN DISTRICT 48.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 56401 57600 -f1
/flx8/flat_048/district_048_1.dat

GEN DISTRICT 49.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 57601 58800 -f1
/flx8/flat_049/district_049_1.dat

GEN DISTRICT 4.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 3601 4800 -f1
/flx8/flat_004/district_004_1.dat

GEN DISTRICT 50.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 58801 60000 -f1
/flx8/flat_050/district_050_1.dat

GEN DISTRICT 51.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 60001 61200 -f1
/flx8/flat_051/district_051_1.dat

GEN DISTRICT 52.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 61201 62400 -f1
/flx8/flat_052/district_052_1.dat

GEN DISTRICT 53.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 62401 63600 -f1
/flx8/flat_053/district_053_1.dat

GEN DISTRICT 54.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 63601 64800 -f1
/flx8/flat_054/district_054_1.dat

GEN DISTRICT 55.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 64801 66000 -f1
/flx8/flat_055/district_055_1.dat

GEN DISTRICT 56.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 66001 67200 -f1
/flx8/flat_056/district_056_1.dat

GEN DISTRICT 57.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 67201 68400 -f1
/flx8/flat_057/district_057_1.dat

GEN DISTRICT 58.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 68401 69600 -f1
/flx8/flat_058/district_058_1.dat

GEN DISTRICT 59.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 69601 70800 -f1
/flx8/flat_059/district_059_1.dat

GEN DISTRICT 5.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 4801 6000 -f1
/flx8/flat_005/district_005_1.dat

GEN DISTRICT 60.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 70801 72000 -f1
/flx8/flat_060/district_060_1.dat

GEN DISTRICT 61.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 72001 73200 -f1
/flx8/flat_061/district_061_1.dat

GEN DISTRICT 62.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 73201 74400 -f1
/flx8/flat_062/district_062_1.dat

GEN DISTRICT 63.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 74401 75600 -f1
/flx8/flat_063/district_063_1.dat

GEN DISTRICT 64.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 75601 76800 -f1
/flx8/flat_064/district_064_1.dat

GEN DISTRICT 65.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 76801 78000 -f1
/flx8/flat_065/district_065_1.dat

GEN DISTRICT 66.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 78001 79200 -f1
/flx8/flat_066/district_066_1.dat

GEN DISTRICT 67.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 79201 80400 -f1
/flx8/flat_067/district_067_1.dat

GEN DISTRICT 68.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 80401 81600 -f1
/flx8/flat_068/district_068_1.dat

GEN DISTRICT 69.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 81601 82800 -f1
/flx8/flat_069/district_069_1.dat

GEN DISTRICT 6.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 6001 7200 -f1
/flx8/flat_006/district_006_1.dat

GEN DISTRICT 70.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 82801 84000 -f1
/flx8/flat_070/district_070_1.dat

GEN DISTRICT 71.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 84001 85200 -f1
/flx8/flat_071/district_071_1.dat

GEN DISTRICT 72.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 85201 86400 -f1
/flx8/flat_072/district_072_1.dat

GEN DISTRICT 73.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 86401 87600 -f1
/flx8/flat_073/district_073_1.dat

GEN DISTRICT 74.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 87601 88800 -f1
/flx8/flat_074/district_074_1.dat

GEN DISTRICT 75.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 88801 90000 -f1
/flx8/flat_075/district_075_1.dat

GEN DISTRICT 76.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 90001 91200 -f1
/flx8/flat_076/district_076_1.dat

GEN DISTRICT 77.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 91201 92400 -f1
/flx8/flat_077/district_077_1.dat

GEN DISTRICT 78.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 92401 93600 -f1
/flx8/flat_078/district_078_1.dat

GEN DISTRICT 79.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 93601 94800 -f1
/flx8/flat_079/district_079_1.dat

GEN DISTRICT 7.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 7201 8400 -f1
/flx8/flat_007/district_007_1.dat

GEN DISTRICT 80.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 94801 96000 -f1
/flx8/flat_080/district_080_1.dat

GEN DISTRICT 8.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 8401 9600 -f1
/flx8/flat_008/district_008_1.dat

GEN DISTRICT 9.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 9601 10800 -f1
/flx8/flat_009/district_009_1.dat

GEN HISTORY 10.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 10801 12000 -f1
/flx8/flat_010/history_010_1.dat

GEN HISTORY 11.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 12001 13200 -f1
/flx8/flat_011/history_011_1.dat

GEN HISTORY 12.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 13201 14400 -f1
/flx8/flat_012/history_012_1.dat

GEN HISTORY 13.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 14401 15600 -f1
/flx8/flat_013/history_013_1.dat

GEN HISTORY 14.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 15601 16800 -f1
/flx8/flat_014/history_014_1.dat

GEN HISTORY 15.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 16801 18000 -f1
/flx8/flat_015/history_015_1.dat

GEN HISTORY 16.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 18001 19200 -f1
/flx8/flat_016/history_016_1.dat

GEN HISTORY 17.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 19201 20400 -f1
/flx8/flat_017/history_017_1.dat

GEN HISTORY 18.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 20401 21600 -f1
/flx8/flat_018/history_018_1.dat

GEN HISTORY 19.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 21601 22800 -f1
/flx8/flat_019/history_019_1.dat

GEN HISTORY 1.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1 1200 -f1
/flx8/flat_001/history_001_1.dat

GEN HISTORY 20.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 22801 24000 -f1
/flx8/flat_020/history_020_1.dat

GEN HISTORY 21.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 24001 25200 -f1
/flx8/flat_021/history_021_1.dat

GEN HISTORY 22.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 25201 26400 -f1
/flx8/flat_022/history_022_1.dat

GEN HISTORY 23.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 26401 27600 -f1
/flx8/flat_023/history_023_1.dat

GEN HISTORY 24.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 27601 28800 -f1
/flx8/flat_024/history_024_1.dat

GEN HISTORY 25.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 28801 30000 -f1
/flx8/flat_025/history_025_1.dat

GEN HISTORY 26.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 30001 31200 -f1
/flx8/flat_026/history_026_1.dat

GEN HISTORY 27.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 31201 32400 -f1
/flx8/flat_027/history_027_1.dat

GEN HISTORY 28.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 32401 33600 -f1
/flx8/flat_028/history_028_1.dat

GEN HISTORY 29.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 33601 34800 -f1
/flx8/flat_029/history_029_1.dat

GEN HISTORY 2.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1201 2400 -f1
/flx8/flat_002/history_002_1.dat

GEN HISTORY 30.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 34801 36000 -f1
/flx8/flat_030/history_030_1.dat

GEN HISTORY 31.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 36001 37200 -f1
/flx8/flat_031/history_031_1.dat

GEN HISTORY 32.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 37201 38400 -f1
/flx8/flat_032/history_032_1.dat

GEN HISTORY 33.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 38401 39600 -f1
/flx8/flat_033/history_033_1.dat

GEN HISTORY 34.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 39601 40800 -f1
/flx8/flat_034/history_034_1.dat

GEN HISTORY 35.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 40801 42000 -f1
/flx8/flat_035/history_035_1.dat

GEN HISTORY 36.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 42001 43200 -f1
/flx8/flat_036/history_036_1.dat

GEN HISTORY 37.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 43201 44400 -f1
/flx8/flat_037/history_037_1.dat

GEN HISTORY 38.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 44401 45600 -f1
/flx8/flat_038/history_038_1.dat

GEN HISTORY 39.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 45601 46800 -f1
/flx8/flat_039/history_039_1.dat

GEN HISTORY 3.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 2401 3600 -f1
/flx8/flat_003/history_003_1.dat

GEN HISTORY 40.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 46801 48000 -f1
/flx8/flat_040/history_040_1.dat

GEN HISTORY 41.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 48001 49200 -f1
/flx8/flat_041/history_041_1.dat

GEN HISTORY 42.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 49201 50400 -f1
/flx8/flat_042/history_042_1.dat

GEN HISTORY 43.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 50401 51600 -f1
/flx8/flat_043/history_043_1.dat

GEN HISTORY 44.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 51601 52800 -f1
/flx8/flat_044/history_044_1.dat

GEN HISTORY 45.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 52801 54000 -f1
/flx8/flat_045/history_045_1.dat

GEN HISTORY 46.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 54001 55200 -f1
/flx8/flat_046/history_046_1.dat

GEN HISTORY 47.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 55201 56400 -f1
/flx8/flat_047/history_047_1.dat

GEN HISTORY 48.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 56401 57600 -f1
/flx8/flat_048/history_048_1.dat

GEN HISTORY 49.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 57601 58800 -f1
/flx8/flat_049/history_049_1.dat

GEN HISTORY 4.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 3601 4800 -f1
/flx8/flat_004/history_004_1.dat

GEN HISTORY 50.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 58801 60000 -f1
/flx8/flat_050/history_050_1.dat

GEN HISTORY 51.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 60001 61200 -f1
/flx8/flat_051/history_051_1.dat

GEN HISTORY 52.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 61201 62400 -f1
/flx8/flat_052/history_052_1.dat

GEN HISTORY 53.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 62401 63600 -f1
/flx8/flat_053/history_053_1.dat

GEN HISTORY 54.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 63601 64800 -f1
/flx8/flat_054/history_054_1.dat

GEN HISTORY 55.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 64801 66000 -f1
/flx8/flat_055/history_055_1.dat

GEN HISTORY 56.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 66001 67200 -f1
/flx8/flat_056/history_056_1.dat

GEN HISTORY 57.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 67201 68400 -f1
/flx8/flat_057/history_057_1.dat

GEN HISTORY 58.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 68401 69600 -f1
/flx8/flat_058/history_058_1.dat

GEN HISTORY 59.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 69601 70800 -f1
/flx8/flat_059/history_059_1.dat

GEN HISTORY 5.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 4801 6000 -f1
/flx8/flat_005/history_005_1.dat

GEN HISTORY 60.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 70801 72000 -f1
/flx8/flat_060/history_060_1.dat

GEN HISTORY 61.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 72001 73200 -f1
/flx8/flat_061/history_061_1.dat

GEN HISTORY 62.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 73201 74400 -f1
/flx8/flat_062/history_062_1.dat

GEN HISTORY 63.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 74401 75600 -f1
/flx8/flat_063/history_063_1.dat

GEN HISTORY 64.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 75601 76800 -f1
/flx8/flat_064/history_064_1.dat

GEN HISTORY 65.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 76801 78000 -f1
/flx8/flat_065/history_065_1.dat

GEN HISTORY 66.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 78001 79200 -f1
/flx8/flat_066/history_066_1.dat

GEN HISTORY 67.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 79201 80400 -f1
/flx8/flat_067/history_067_1.dat

GEN HISTORY 68.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 80401 81600 -f1
/flx8/flat_068/history_068_1.dat

GEN HISTORY 69.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 81601 82800 -f1
/flx8/flat_069/history_069_1.dat

GEN HISTORY 6.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 6001 7200 -f1
/flx8/flat_006/history_006_1.dat

GEN HISTORY 70.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 82801 84000 -f1
/flx8/flat_070/history_070_1.dat

GEN HISTORY 71.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 84001 85200 -f1
/flx8/flat_071/history_071_1.dat

GEN HISTORY 72.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 85201 86400 -f1
/flx8/flat_072/history_072_1.dat

GEN HISTORY 73.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 86401 87600 -f1
/flx8/flat_073/history_073_1.dat

GEN HISTORY 74.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 87601 88800 -f1
/flx8/flat_074/history_074_1.dat

GEN HISTORY 75.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 88801 90000 -f1
/flx8/flat_075/history_075_1.dat

GEN HISTORY 76.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 90001 91200 -f1
/flx8/flat_076/history_076_1.dat

GEN HISTORY 77.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 91201 92400 -f1
/flx8/flat_077/history_077_1.dat

GEN HISTORY 78.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 92401 93600 -f1
/flx8/flat_078/history_078_1.dat

GEN HISTORY 79.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 93601 94800 -f1
/flx8/flat_079/history_079_1.dat

GEN HISTORY 7.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 7201 8400 -f1
/flx8/flat_007/history_007_1.dat

GEN HISTORY 80.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 94801 96000 -f1
/flx8/flat_080/history_080_1.dat

GEN HISTORY 8.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 8401 9600 -f1
/flx8/flat_008/history_008_1.dat

GEN HISTORY 9.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 9601 10800 -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 10801 12000 -f1
/flx8/flat_010/neworder_010_1.dat

GEN NEW ORDER 11.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 12001 13200 -f1
/flx8/flat_011/neworder_011_1.dat

GEN NEW ORDER 12.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 13201 14400 -f1
/flx8/flat_012/neworder_012_1.dat

GEN NEW ORDER 13.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 14401 15600 -f1
/flx8/flat_013/neworder_013_1.dat

GEN NEW ORDER 14.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 15601 16800 -f1
/flx8/flat_014/neworder_014_1.dat

GEN NEW ORDER 15.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 16801 18000 -f1
/flx8/flat_015/neworder_015_1.dat

GEN NEW ORDER 16.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 18001 19200 -f1
/flx8/flat_016/neworder_016_1.dat

GEN NEW ORDER 17.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 19201 20400 -f1
/flx8/flat_017/neworder_017_1.dat

GEN NEW ORDER 18.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 20401 21600 -f1
/flx8/flat_018/neworder_018_1.dat

GEN NEW ORDER 19.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 21601 22800 -f1
/flx8/flat_019/neworder_019_1.dat

GEN NEW ORDER 1.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1 1200 -f1
/flx8/flat_001/neworder_001_1.dat

GEN NEW ORDER 20.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 22801 24000 -f1
/flx8/flat_020/neworder_020_1.dat

GEN NEW ORDER 21.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 24001 25200 -f1
/flx8/flat_021/neworder_021_1.dat

GEN NEW ORDER 22.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 25201 26400 -f1
/flx8/flat_022/neworder_022_1.dat

GEN NEW ORDER 23.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 26401 27600 -f1
/flx8/flat_023/neworder_023_1.dat

GEN NEW ORDER 24.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 27601 28800 -f1
/flx8/flat_024/neworder_024_1.dat

GEN NEW ORDER 25.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 28801 30000 -f1
/flx8/flat_025/neworder_025_1.dat

GEN NEW ORDER 26.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 30001 31200 -f1
/flx8/flat_026/neworder_026_1.dat

GEN NEW ORDER 27.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 31201 32400 -f1
/flx8/flat_027/neworder_027_1.dat

GEN NEW ORDER 28.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 32401 33600 -f1
/flx8/flat_028/neworder_028_1.dat

GEN NEW ORDER 29.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 33601 34800 -f1
/flx8/flat_029/neworder_029_1.dat

GEN NEW ORDER 2.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1201 2400 -f1
/flx8/flat_002/neworder_002_1.dat

GEN NEW ORDER 30.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 34801 36000 -f1
/flx8/flat_030/neworder_030_1.dat

GEN NEW ORDER 31.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 36001 37200 -f1
/flx8/flat_031/neworder_031_1.dat

GEN NEW ORDER 32.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 37201 38400 -f1
/flx8/flat_032/neworder_032_1.dat

GEN NEW ORDER 33.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 38401 39600 -f1
/flx8/flat_033/neworder_033_1.dat

GEN NEW ORDER 34.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 39601 40800 -f1
/flx8/flat_034/neworder_034_1.dat

GEN NEW ORDER 35.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 40801 42000 -f1
/flx8/flat_035/neworder_035_1.dat

GEN NEW ORDER 36.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 42001 43200 -f1
/flx8/flat_036/neworder_036_1.dat

GEN NEW ORDER 37.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 43201 44400 -f1
/flx8/flat_037/neworder_037_1.dat

GEN NEW ORDER 38.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 44401 45600 -f1
/flx8/flat_038/neworder_038_1.dat

GEN NEW ORDER 39.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 45601 46800 -f1
/flx8/flat_039/neworder_039_1.dat

GEN NEW ORDER 3.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 2401 3600 -f1
/flx8/flat_003/neworder_003_1.dat

GEN NEW ORDER 40.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 46801 48000 -f1
/flx8/flat_040/neworder_040_1.dat

GEN NEW ORDER 41.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 48001 49200 -f1
/flx8/flat_041/neworder_041_1.dat

GEN NEW ORDER 42.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 49201 50400 -f1
/flx8/flat_042/neworder_042_1.dat

GEN NEW ORDER 43.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 50401 51600 -f1
/flx8/flat_043/neworder_043_1.dat

GEN NEW ORDER 44.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 51601 52800 -f1
/flx8/flat_044/neworder_044_1.dat

GEN NEW ORDER 45.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 52801 54000 -f1
/flx8/flat_045/neworder_045_1.dat

GEN NEW ORDER 46.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 54001 55200 -f1
/flx8/flat_046/neworder_046_1.dat

GEN NEW ORDER 47.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 55201 56400 -f1
/flx8/flat_047/neworder_047_1.dat

GEN NEW ORDER 48.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 56401 57600 -f1
/flx8/flat_048/neworder_048_1.dat

GEN NEW ORDER 49.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 57601 58800 -f1
/flx8/flat_049/neworder_049_1.dat

GEN NEW ORDER 4.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 3601 4800 -f1
/flx8/flat_004/neworder_004_1.dat

GEN NEW ORDER 50.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 58801 60000 -f1
/flx8/flat_050/neworder_050_1.dat

GEN NEW ORDER 51.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 60001 61200 -f1
/flx8/flat_051/neworder_051_1.dat

GEN NEW ORDER 52.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 61201 62400 -f1
/flx8/flat_052/neworder_052_1.dat

GEN NEW ORDER 53.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 62401 63600 -f1
/flx8/flat_053/neworder_053_1.dat

GEN NEW ORDER 54.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 63601 64800 -f1
/flx8/flat_054/neworder_054_1.dat

GEN NEW ORDER 55.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 64801 66000 -f1
/flx8/flat_055/neworder_055_1.dat

GEN NEW ORDER 56.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 66001 67200 -f1
/flx8/flat_056/neworder_056_1.dat

GEN NEW ORDER 57.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 67201 68400 -f1
/flx8/flat_057/neworder_057_1.dat

GEN NEW ORDER 58.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 68401 69600 -f1
/flx8/flat_058/neworder_058_1.dat

GEN NEW ORDER 59.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 69601 70800 -f1
/flx8/flat_059/neworder_059_1.dat

GEN NEW ORDER 5.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 4801 6000 -f1
/flx8/flat_005/neworder_005_1.dat

GEN NEW ORDER 60.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 70801 72000 -f1
/flx8/flat_060/neworder_060_1.dat

GEN NEW ORDER 61.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 72001 73200 -f1
/flx8/flat_061/neworder_061_1.dat

GEN NEW ORDER 62.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 73201 74400 -f1
/flx8/flat_062/neworder_062_1.dat

GEN NEW ORDER 63.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 74401 75600 -f1
/flx8/flat_063/neworder_063_1.dat

GEN NEW ORDER 64.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 75601 76800 -f1
/flx8/flat_064/neworder_064_1.dat

GEN NEW ORDER 65.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 76801 78000 -f1
/flx8/flat_065/neworder_065_1.dat

GEN NEW ORDER 66.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 78001 79200 -f1
/flx8/flat_066/neworder_066_1.dat

GEN NEW ORDER 67.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 79201 80400 -f1
/flx8/flat_067/neworder_067_1.dat

GEN NEW ORDER 68.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 80401 81600 -f1
/flx8/flat_068/neworder_068_1.dat

GEN NEW ORDER 69.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 81601 82800 -f1
/flx8/flat_069/neworder_069_1.dat

GEN NEW ORDER 6.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 6001 7200 -f1
/flx8/flat_006/neworder_006_1.dat

GEN NEW ORDER 70.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 82801 84000 -f1
/flx8/flat_070/neworder_070_1.dat

GEN NEW ORDER 71.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 84001 85200 -f1
/flx8/flat_071/neworder_071_1.dat

GEN NEW ORDER 72.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 85201 86400 -f1
/flx8/flat_072/neworder_072_1.dat

GEN NEW ORDER 73.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 86401 87600 -f1
/flx8/flat_073/neworder_073_1.dat

GEN NEW ORDER 74.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 87601 88800 -f1
/flx8/flat_074/neworder_074_1.dat

GEN NEW ORDER 75.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 88801 90000 -f1
/flx8/flat_075/neworder_075_1.dat

GEN NEW ORDER 76.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 90001 91200 -f1
/flx8/flat_076/neworder_076_1.dat

GEN NEW ORDER 77.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 91201 92400 -f1
/flx8/flat_077/neworder_077_1.dat

GEN NEW ORDER 78.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 92401 93600 -f1
/flx8/flat_078/neworder_078_1.dat

GEN NEW ORDER 79.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 93601 94800 -f1
/flx8/flat_079/neworder_079_1.dat

GEN NEW ORDER 7.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 7201 8400 -f1
/flx8/flat_007/neworder_007_1.dat

GEN NEW ORDER 80.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 94801 96000 -f1
/flx8/flat_080/neworder_080_1.dat

GEN NEW ORDER 8.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 8401 9600 -f1
/flx8/flat_008/neworder_008_1.dat

GEN NEW ORDER 9.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 9601 10800 -f1
/flx8/flat_009/neworder_009_1.dat

GEN ORDERS 10.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10801 12000 -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 12001 13200 -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 13201 14400 -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 14401 15600 -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 15601 16800 -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 16801 18000 -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 18001 19200 -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 19201 20400 -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 20401 21600 -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 21601 22800 -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 1200 -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 22801 24000 -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 24001 25200 -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 25201 26400 -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 26401 27600 -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 27601 28800 -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 28801 30000 -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 30001 31200 -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 31201 32400 -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 32401 33600 -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 33601 34800 -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 1201 2400 -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 34801 36000 -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 36001 37200 -f1
/flx8/flat_031/orders_031_1.dat -f2
/flx8/flat_031/orderline_031_1.dat

GEN ORDERS 32.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37201 38400 -f1
/flx8/flat_032/orders_032_1.dat -f2
/flx8/flat_032/orderline_032_1.dat

GEN ORDERS 33.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38401 39600 -f1
/flx8/flat_033/orders_033_1.dat -f2
/flx8/flat_033/orderline_033_1.dat

GEN ORDERS 34.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39601 40800 -f1
/flx8/flat_034/orders_034_1.dat -f2
/flx8/flat_034/orderline_034_1.dat

GEN ORDERS 35.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40801 42000 -f1
/flx8/flat_035/orders_035_1.dat -f2
/flx8/flat_035/orderline_035_1.dat

GEN ORDERS 36.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 42001 43200 -f1
/flx8/flat_036/orders_036_1.dat -f2
/flx8/flat_036/orderline_036_1.dat

GEN ORDERS 37.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 43201 44400 -f1
/flx8/flat_037/orders_037_1.dat -f2
/flx8/flat_037/orderline_037_1.dat

GEN ORDERS 38.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 44401 45600 -f1
/flx8/flat_038/orders_038_1.dat -f2
/flx8/flat_038/orderline_038_1.dat

GEN ORDERS 39.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 45601 46800 -f1
/flx8/flat_039/orders_039_1.dat -f2
/flx8/flat_039/orderline_039_1.dat

GEN ORDERS 3.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2401 3600 -f1
/flx8/flat_003/orders_003_1.dat -f2
/flx8/flat_003/orderline_003_1.dat

GEN ORDERS 40.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 46801 48000 -f1
/flx8/flat_040/orders_040_1.dat -f2
/flx8/flat_040/orderline_040_1.dat

GEN ORDERS 41.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 48001 49200 -f1
/flx8/flat_041/orders_041_1.dat -f2
/flx8/flat_041/orderline_041_1.dat

GEN ORDERS 42.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 49201 50400 -f1
/flx8/flat_042/orders_042_1.dat -f2
/flx8/flat_042/orderline_042_1.dat

GEN ORDERS 43.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 50401 51600 -f1
/flx8/flat_043/orders_043_1.dat -f2
/flx8/flat_043/orderline_043_1.dat

GEN ORDERS 44.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 51601 52800 -f1
/flx8/flat_044/orders_044_1.dat -f2
/flx8/flat_044/orderline_044_1.dat

GEN ORDERS 45.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 52801 54000 -f1
/flx8/flat_045/orders_045_1.dat -f2
/flx8/flat_045/orderline_045_1.dat

GEN ORDERS 46.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 54001 55200 -f1
/flx8/flat_046/orders_046_1.dat -f2
/flx8/flat_046/orderline_046_1.dat

GEN ORDERS 47.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 55201 56400 -f1
/flx8/flat_047/orders_047_1.dat -f2
/flx8/flat_047/orderline_047_1.dat

GEN ORDERS 48.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 56401 57600 -f1
/flx8/flat_048/orders_048_1.dat -f2
/flx8/flat_048/orderline_048_1.dat

GEN ORDERS 49.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 57601 58800 -f1
/flx8/flat_049/orders_049_1.dat -f2
/flx8/flat_049/orderline_049_1.dat

GEN ORDERS 4.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3601 4800 -f1
/flx8/flat_004/orders_004_1.dat -f2
/flx8/flat_004/orderline_004_1.dat

GEN ORDERS 50.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 58801 60000 -f1
/flx8/flat_050/orders_050_1.dat -f2
/flx8/flat_050/orderline_050_1.dat

GEN ORDERS 51.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 60001 61200 -f1
/flx8/flat_051/orders_051_1.dat -f2
/flx8/flat_051/orderline_051_1.dat

GEN ORDERS 52.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 61201 62400 -f1
/flx8/flat_052/orders_052_1.dat -f2
/flx8/flat_052/orderline_052_1.dat

GEN ORDERS 53.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 62401 63600 -f1
/flx8/flat_053/orders_053_1.dat -f2
/flx8/flat_053/orderline_053_1.dat

GEN ORDERS 54.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 63601 64800 -f1
/flx8/flat_054/orders_054_1.dat -f2
/flx8/flat_054/orderline_054_1.dat

GEN ORDERS 55.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 64801 66000 -f1
/flx8/flat_055/orders_055_1.dat -f2
/flx8/flat_055/orderline_055_1.dat

GEN ORDERS 56.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 66001 67200 -f1
/flx8/flat_056/orders_056_1.dat -f2
/flx8/flat_056/orderline_056_1.dat

GEN ORDERS 57.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 67201 68400 -f1
/flx8/flat_057/orders_057_1.dat -f2
/flx8/flat_057/orderline_057_1.dat

GEN ORDERS 58.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 68401 69600 -f1
/flx8/flat_058/orders_058_1.dat -f2
/flx8/flat_058/orderline_058_1.dat

GEN ORDERS 59.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 69601 70800 -f1
/flx8/flat_059/orders_059_1.dat -f2
/flx8/flat_059/orderline_059_1.dat

GEN ORDERS 5.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4801 6000 -f1
/flx8/flat_005/orders_005_1.dat -f2
/flx8/flat_005/orderline_005_1.dat

GEN ORDERS 60.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 70801 72000 -f1
/flx8/flat_060/orders_060_1.dat -f2
/flx8/flat_060/orderline_060_1.dat

GEN ORDERS 61.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 72001 73200 -f1
/flx8/flat_061/orders_061_1.dat -f2
/flx8/flat_061/orderline_061_1.dat

GEN ORDERS 62.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 73201 74400 -f1
/flx8/flat_062/orders_062_1.dat -f2
/flx8/flat_062/orderline_062_1.dat

GEN ORDERS 63.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 74401 75600 -f1
/flx8/flat_063/orders_063_1.dat -f2
/flx8/flat_063/orderline_063_1.dat

GEN ORDERS 64.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 75601 76800 -f1
/flx8/flat_064/orders_064_1.dat -f2
/flx8/flat_064/orderline_064_1.dat

GEN ORDERS 65.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 76801 78000 -f1
/flx8/flat_065/orders_065_1.dat -f2
/flx8/flat_065/orderline_065_1.dat

GEN ORDERS 66.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 78001 79200 -f1
/flx8/flat_066/orders_066_1.dat -f2
/flx8/flat_066/orderline_066_1.dat

GEN ORDERS 67.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 79201 80400 -f1
/flx8/flat_067/orders_067_1.dat -f2
/flx8/flat_067/orderline_067_1.dat

GEN ORDERS 68.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 80401 81600 -f1
/flx8/flat_068/orders_068_1.dat -f2
/flx8/flat_068/orderline_068_1.dat

GEN ORDERS 69.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 81601 82800 -f1
/flx8/flat_069/orders_069_1.dat -f2
/flx8/flat_069/orderline_069_1.dat

GEN ORDERS 6.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6001 7200 -f1
/flx8/flat_006/orders_006_1.dat -f2
/flx8/flat_006/orderline_006_1.dat

GEN ORDERS 70.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 82801 84000 -f1
/flx8/flat_070/orders_070_1.dat -f2
/flx8/flat_070/orderline_070_1.dat

GEN ORDERS 71.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 84001 85200 -f1
/flx8/flat_071/orders_071_1.dat -f2
/flx8/flat_071/orderline_071_1.dat

GEN ORDERS 72.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 85201 86400 -f1
/flx8/flat_072/orders_072_1.dat -f2
/flx8/flat_072/orderline_072_1.dat

GEN ORDERS 73.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 86401 87600 -f1
/flx8/flat_073/orders_073_1.dat -f2
/flx8/flat_073/orderline_073_1.dat

GEN ORDERS 74.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 87601 88800 -f1
/flx8/flat_074/orders_074_1.dat -f2
/flx8/flat_074/orderline_074_1.dat

GEN ORDERS 75.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 88801 90000 -f1
/flx8/flat_075/orders_075_1.dat -f2
/flx8/flat_075/orderline_075_1.dat

GEN ORDERS 76.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 90001 91200 -f1
/flx8/flat_076/orders_076_1.dat -f2
/flx8/flat_076/orderline_076_1.dat

GEN ORDERS 77.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 91201 92400 -f1
/flx8/flat_077/orders_077_1.dat -f2
/flx8/flat_077/orderline_077_1.dat

GEN ORDERS 78.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 92401 93600 -f1
/flx8/flat_078/orders_078_1.dat -f2
/flx8/flat_078/orderline_078_1.dat

GEN ORDERS 79.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 93601 94800 -f1
/flx8/flat_079/orders_079_1.dat -f2
/flx8/flat_079/orderline_079_1.dat

GEN ORDERS 7.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7201 8400 -f1
/flx8/flat_007/orders_007_1.dat -f2
/flx8/flat_007/orderline_007_1.dat

GEN ORDERS 80.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 94801 96000 -f1
/flx8/flat_080/orders_080_1.dat -f2
/flx8/flat_080/orderline_080_1.dat

GEN ORDERS 8.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8401 9600 -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 9601 10800 -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 10801 12000 -f1
/flx8/flat_010/stock_010_1.dat

GEN STOCK 11.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12001 13200 -f1
/flx8/flat_011/stock_011_1.dat

GEN STOCK 12.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13201 14400 -f1
/flx8/flat_012/stock_012_1.dat

GEN STOCK 13.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14401 15600 -f1
/flx8/flat_013/stock_013_1.dat

GEN STOCK 14.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15601 16800 -f1
/flx8/flat_014/stock_014_1.dat

GEN STOCK 15.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16801 18000 -f1
/flx8/flat_015/stock_015_1.dat

GEN STOCK 16.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18001 19200 -f1
/flx8/flat_016/stock_016_1.dat

GEN STOCK 17.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19201 20400 -f1
/flx8/flat_017/stock_017_1.dat

GEN STOCK 18.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20401 21600 -f1
/flx8/flat_018/stock_018_1.dat

GEN STOCK 19.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21601 22800 -f1
/flx8/flat_019/stock_019_1.dat

GEN STOCK 1.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1 1200 -f1
/flx8/flat_001/stock_001_1.dat

GEN STOCK 20.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22801 24000 -f1
/flx8/flat_020/stock_020_1.dat

GEN STOCK 21.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24001 25200 -f1
/flx8/flat_021/stock_021_1.dat

GEN STOCK 22.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 25201 26400 -f1
/flx8/flat_022/stock_022_1.dat

GEN STOCK 23.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26401 27600 -f1
/flx8/flat_023/stock_023_1.dat

GEN STOCK 24.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27601 28800 -f1
/flx8/flat_024/stock_024_1.dat

GEN STOCK 25.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28801 30000 -f1
/flx8/flat_025/stock_025_1.dat

GEN STOCK 26.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30001 31200 -f1
/flx8/flat_026/stock_026_1.dat

GEN STOCK 27.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31201 32400 -f1
/flx8/flat_027/stock_027_1.dat

GEN STOCK 28.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32401 33600 -f1
/flx8/flat_028/stock_028_1.dat

GEN STOCK 29.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33601 34800 -f1
/flx8/flat_029/stock_029_1.dat

GEN STOCK 2.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1201 2400 -f1
/flx8/flat_002/stock_002_1.dat

GEN STOCK 30.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34801 36000 -f1
/flx8/flat_030/stock_030_1.dat

GEN STOCK 31.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36001 37200 -f1
/flx8/flat_031/stock_031_1.dat

GEN STOCK 32.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 37201 38400 -f1
/flx8/flat_032/stock_032_1.dat

GEN STOCK 33.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38401 39600 -f1
/flx8/flat_033/stock_033_1.dat

GEN STOCK 34.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39601 40800 -f1
/flx8/flat_034/stock_034_1.dat

GEN STOCK 35.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40801 42000 -f1
/flx8/flat_035/stock_035_1.dat

GEN STOCK 36.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 42001 43200 -f1
/flx8/flat_036/stock_036_1.dat

GEN STOCK 37.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 43201 44400 -f1
/flx8/flat_037/stock_037_1.dat

GEN STOCK 38.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 44401 45600 -f1
/flx8/flat_038/stock_038_1.dat

GEN STOCK 39.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 45601 46800 -f1
/flx8/flat_039/stock_039_1.dat

GEN STOCK 3.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2401 3600 -f1
/flx8/flat_003/stock_003_1.dat

GEN STOCK 40.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 46801 48000 -f1
/flx8/flat_040/stock_040_1.dat

GEN STOCK 41.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 48001 49200 -f1
/flx8/flat_041/stock_041_1.dat

GEN STOCK 42.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 49201 50400 -f1
/flx8/flat_042/stock_042_1.dat

GEN STOCK 43.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 50401 51600 -f1
/flx8/flat_043/stock_043_1.dat

GEN STOCK 44.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 51601 52800 -f1
/flx8/flat_044/stock_044_1.dat

GEN STOCK 45.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 52801 54000 -f1
/flx8/flat_045/stock_045_1.dat

GEN STOCK 46.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 54001 55200 -f1
/flx8/flat_046/stock_046_1.dat

GEN STOCK 47.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 55201 56400 -f1
/flx8/flat_047/stock_047_1.dat

GEN STOCK 48.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 56401 57600 -f1
/flx8/flat_048/stock_048_1.dat

GEN STOCK 49.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 57601 58800 -f1
/flx8/flat_049/stock_049_1.dat

GEN STOCK 4.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3601 4800 -f1
/flx8/flat_004/stock_004_1.dat

GEN STOCK 50.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 58801 60000 -f1
/flx8/flat_050/stock_050_1.dat

GEN STOCK 51.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 60001 61200 -f1
/flx8/flat_051/stock_051_1.dat

GEN STOCK 52.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 61201 62400 -f1
/flx8/flat_052/stock_052_1.dat

GEN STOCK 53.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 62401 63600 -f1
/flx8/flat_053/stock_053_1.dat

GEN STOCK 54.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 63601 64800 -f1
/flx8/flat_054/stock_054_1.dat

GEN STOCK 55.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 64801 66000 -f1
/flx8/flat_055/stock_055_1.dat

GEN STOCK 56.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 66001 67200 -f1
/flx8/flat_056/stock_056_1.dat

GEN STOCK 57.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 67201 68400 -f1
/flx8/flat_057/stock_057_1.dat

GEN STOCK 58.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 68401 69600 -f1
/flx8/flat_058/stock_058_1.dat

GEN STOCK 59.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 69601 70800 -f1
/flx8/flat_059/stock_059_1.dat

GEN STOCK 5.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4801 6000 -f1
/flx8/flat_005/stock_005_1.dat

GEN STOCK 60.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 70801 72000 -f1
/flx8/flat_060/stock_060_1.dat

GEN STOCK 61.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 72001 73200 -f1
/flx8/flat_061/stock_061_1.dat

GEN STOCK 62.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 73201 74400 -f1
/flx8/flat_062/stock_062_1.dat

GEN STOCK 63.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 74401 75600 -f1
/flx8/flat_063/stock_063_1.dat

GEN STOCK 64.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 75601 76800 -f1
/flx8/flat_064/stock_064_1.dat

GEN STOCK 65.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 76801 78000 -f1
/flx8/flat_065/stock_065_1.dat

GEN STOCK 66.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 78001 79200 -f1
/flx8/flat_066/stock_066_1.dat

GEN STOCK 67.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 79201 80400 -f1
/flx8/flat_067/stock_067_1.dat

GEN STOCK 68.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 80401 81600 -f1
/flx8/flat_068/stock_068_1.dat

GEN STOCK 69.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 81601 82800 -f1
/flx8/flat_069/stock_069_1.dat

GEN STOCK 6.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6001 7200 -f1
/flx8/flat_006/stock_006_1.dat

GEN STOCK 70.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 82801 84000 -f1
/flx8/flat_070/stock_070_1.dat

GEN STOCK 71.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 84001 85200 -f1
/flx8/flat_071/stock_071_1.dat

GEN STOCK 72.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 85201 86400 -f1
/flx8/flat_072/stock_072_1.dat

GEN STOCK 73.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 86401 87600 -f1
/flx8/flat_073/stock_073_1.dat

GEN STOCK 74.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 87601 88800 -f1
/flx8/flat_074/stock_074_1.dat

GEN STOCK 75.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 88801 90000 -f1
/flx8/flat_075/stock_075_1.dat

GEN STOCK 76.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 90001 91200 -f1
/flx8/flat_076/stock_076_1.dat

GEN STOCK 77.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 91201 92400 -f1
/flx8/flat_077/stock_077_1.dat

GEN STOCK 78.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 92401 93600 -f1
/flx8/flat_078/stock_078_1.dat

GEN STOCK 79.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 93601 94800 -f1
/flx8/flat_079/stock_079_1.dat

GEN STOCK 7.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7201 8400 -f1
/flx8/flat_007/stock_007_1.dat

GEN STOCK 80.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 94801 96000 -f1
/flx8/flat_080/stock_080_1.dat

GEN STOCK 8.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8401 9600 -f1
/flx8/flat_008/stock_008_1.dat

GEN STOCK 9.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9601 10800 -f1
/flx8/flat_009/stock_009_1.dat

GEN WAREHOUSE 10.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 10801 12000 -f1
/flx8/flat_010/warehouse_010_1.dat

GEN WAREHOUSE 11.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 12001 13200 -f1
/flx8/flat_011/warehouse_011_1.dat

GEN WAREHOUSE 12.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 13201 14400 -f1
/flx8/flat_012/warehouse_012_1.dat

GEN WAREHOUSE 13.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 14401 15600 -f1
/flx8/flat_013/warehouse_013_1.dat

GEN WAREHOUSE 14.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 15601 16800 -f1
/flx8/flat_014/warehouse_014_1.dat

GEN WAREHOUSE 15.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 16801 18000 -f1
/flx8/flat_015/warehouse_015_1.dat

GEN WAREHOUSE 16.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 18001 19200 -f1
/flx8/flat_016/warehouse_016_1.dat

GEN WAREHOUSE 17.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 19201 20400 -f1
/flx8/flat_017/warehouse_017_1.dat

GEN WAREHOUSE 18.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 20401 21600 -f1
/flx8/flat_018/warehouse_018_1.dat

GEN WAREHOUSE 19.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 21601 22800 -f1
/flx8/flat_019/warehouse_019_1.dat

GEN WAREHOUSE 1.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1 1200 -f1
/flx8/flat_001/warehouse_001_1.dat

GEN WAREHOUSE 20.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 22801 24000 -f1
/flx8/flat_020/warehouse_020_1.dat

GEN WAREHOUSE 21.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 24001 25200 -f1
/flx8/flat_021/warehouse_021_1.dat

GEN WAREHOUSE 22.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 25201 26400 -f1
/flx8/flat_022/warehouse_022_1.dat

GEN WAREHOUSE 23.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 26401 27600 -f1
/flx8/flat_023/warehouse_023_1.dat

GEN WAREHOUSE 24.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 27601 28800 -f1
/flx8/flat_024/warehouse_024_1.dat

GEN WAREHOUSE 25.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 28801 30000 -f1
/flx8/flat_025/warehouse_025_1.dat

GEN WAREHOUSE 26.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 30001 31200 -f1
/flx8/flat_026/warehouse_026_1.dat

GEN WAREHOUSE 27.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 31201 32400 -f1
/flx8/flat_027/warehouse_027_1.dat

GEN WAREHOUSE 28.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 32401 33600 -f1
/flx8/flat_028/warehouse_028_1.dat

GEN WAREHOUSE 29.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 33601 34800 -f1
/flx8/flat_029/warehouse_029_1.dat

GEN WAREHOUSE 2.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1201 2400 -f1
/flx8/flat_002/warehouse_002_1.dat

GEN WAREHOUSE 30.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 34801 36000 -f1
/flx8/flat_030/warehouse_030_1.dat

GEN WAREHOUSE 31.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 36001 37200 -f1
/flx8/flat_031/warehouse_031_1.dat

GEN WAREHOUSE 32.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 37201 38400 -f1
/flx8/flat_032/warehouse_032_1.dat

GEN WAREHOUSE 33.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 38401 39600 -f1
/flx8/flat_033/warehouse_033_1.dat

GEN WAREHOUSE 34.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 39601 40800 -f1
/flx8/flat_034/warehouse_034_1.dat

GEN WAREHOUSE 35.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 40801 42000 -f1
/flx8/flat_035/warehouse_035_1.dat

GEN WAREHOUSE 36.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 42001 43200 -f1
/flx8/flat_036/warehouse_036_1.dat

GEN WAREHOUSE 37.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 43201 44400 -f1
/flx8/flat_037/warehouse_037_1.dat

GEN WAREHOUSE 38.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 44401 45600 -f1
/flx8/flat_038/warehouse_038_1.dat

GEN WAREHOUSE 39.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 45601 46800 -f1
/flx8/flat_039/warehouse_039_1.dat

GEN WAREHOUSE 3.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 2401 3600 -f1
/flx8/flat_003/warehouse_003_1.dat

GEN WAREHOUSE 40.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 46801 48000 -f1
/flx8/flat_040/warehouse_040_1.dat

GEN WAREHOUSE 41.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 48001 49200 -f1
/flx8/flat_041/warehouse_041_1.dat

GEN WAREHOUSE 42.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 49201 50400 -f1
/flx8/flat_042/warehouse_042_1.dat

GEN WAREHOUSE 43.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 50401 51600 -f1
/flx8/flat_043/warehouse_043_1.dat

GEN WAREHOUSE 44.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 51601 52800 -f1
/flx8/flat_044/warehouse_044_1.dat

GEN WAREHOUSE 45.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 52801 54000 -f1
/flx8/flat_045/warehouse_045_1.dat

GEN WAREHOUSE 46.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 54001 55200 -f1
/flx8/flat_046/warehouse_046_1.dat

GEN WAREHOUSE 47.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 55201 56400 -f1
/flx8/flat_047/warehouse_047_1.dat

GEN WAREHOUSE 48.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 56401 57600 -f1
/flx8/flat_048/warehouse_048_1.dat

GEN WAREHOUSE 49.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 57601 58800 -f1
/flx8/flat_049/warehouse_049_1.dat

GEN WAREHOUSE 4.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 3601 4800 -f1
/flx8/flat_004/warehouse_004_1.dat

GEN WAREHOUSE 50.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 58801 60000 -f1
/flx8/flat_050/warehouse_050_1.dat

GEN WAREHOUSE 51.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 60001 61200 -f1
/flx8/flat_051/warehouse_051_1.dat

GEN WAREHOUSE 52.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 61201 62400 -f1
/flx8/flat_052/warehouse_052_1.dat

GEN WAREHOUSE 53.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 62401 63600 -f1
/flx8/flat_053/warehouse_053_1.dat

GEN WAREHOUSE 54.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 63601 64800 -f1
/flx8/flat_054/warehouse_054_1.dat

GEN WAREHOUSE 55.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 64801 66000 -f1
/flx8/flat_055/warehouse_055_1.dat

GEN WAREHOUSE 56.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 66001 67200 -f1
/flx8/flat_056/warehouse_056_1.dat

GEN WAREHOUSE 57.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 67201 68400 -f1
/flx8/flat_057/warehouse_057_1.dat

GEN WAREHOUSE 58.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 68401 69600 -f1
/flx8/flat_058/warehouse_058_1.dat

GEN WAREHOUSE 59.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 69601 70800 -f1
/flx8/flat_059/warehouse_059_1.dat

GEN WAREHOUSE 5.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 4801 6000 -f1
/flx8/flat_005/warehouse_005_1.dat

GEN WAREHOUSE 60.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 70801 72000 -f1
/flx8/flat_060/warehouse_060_1.dat

GEN WAREHOUSE 61.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 72001 73200 -f1
/flx8/flat_061/warehouse_061_1.dat

GEN WAREHOUSE 62.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 73201 74400 -f1
/flx8/flat_062/warehouse_062_1.dat

GEN WAREHOUSE 63.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 74401 75600 -f1
/flx8/flat_063/warehouse_063_1.dat

GEN WAREHOUSE 64.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 75601 76800 -f1
/flx8/flat_064/warehouse_064_1.dat

GEN WAREHOUSE 65.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 76801 78000 -f1
/flx8/flat_065/warehouse_065_1.dat

GEN WAREHOUSE 66.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 78001 79200 -f1
/flx8/flat_066/warehouse_066_1.dat

GEN WAREHOUSE 67.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 79201 80400 -f1
/flx8/flat_067/warehouse_067_1.dat

GEN WAREHOUSE 68.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 80401 81600 -f1
/flx8/flat_068/warehouse_068_1.dat

GEN WAREHOUSE 69.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 81601 82800 -f1
/flx8/flat_069/warehouse_069_1.dat

GEN WAREHOUSE 6.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 6001 7200 -f1  
/flx8/flat_006/warehouse_006_1.dat
```

GEN WAREHOUSE 70.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 82801 84000 -f1  
/flx8/flat_070/warehouse_070_1.dat
```

GEN WAREHOUSE 71.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 84001 85200 -f1  
/flx8/flat_071/warehouse_071_1.dat
```

GEN WAREHOUSE 72.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 85201 86400 -f1  
/flx8/flat_072/warehouse_072_1.dat
```

GEN WAREHOUSE 73.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 86401 87600 -f1  
/flx8/flat_073/warehouse_073_1.dat
```

GEN WAREHOUSE 74.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 87601 88800 -f1  
/flx8/flat_074/warehouse_074_1.dat
```

GEN WAREHOUSE 75.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 88801 90000 -f1  
/flx8/flat_075/warehouse_075_1.dat
```

GEN WAREHOUSE 76.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 90001 91200 -f1  
/flx8/flat_076/warehouse_076_1.dat
```

GEN WAREHOUSE 77.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 91201 92400 -f1  
/flx8/flat_077/warehouse_077_1.dat
```

GEN WAREHOUSE 78.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 92401 93600 -f1  
/flx8/flat_078/warehouse_078_1.dat
```

GEN WAREHOUSE 79.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 93601 94800 -f1  
/flx8/flat_079/warehouse_079_1.dat
```

GEN WAREHOUSE 7.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 7201 8400 -f1  
/flx8/flat_007/warehouse_007_1.dat
```

GEN WAREHOUSE 80.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 94801 96000 -f1  
/flx8/flat_080/warehouse_080_1.dat
```

GEN WAREHOUSE 8.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 8401 9600 -f1  
/flx8/flat_008/warehouse_008_1.dat
```

GEN WAREHOUSE 9.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 9601 10800 -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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 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 36000000 INSERT INTO CUSTOMER32;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER33 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER33 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_033/customer_033_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER33;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER34 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER34 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_034/customer_034_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER34;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER35 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER35 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_035/customer_035_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER35;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER36 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER36 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_036/customer_036_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER36;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER37 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER37 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_037/customer_037_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER37;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER38 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_038/customer_038_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER39 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER39 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_039/customer_039_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER39;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER40 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER40 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_040/customer_040_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER40;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER41 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER41 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_041/customer_041_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER41;
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 36000000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER42 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER42 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_042/customer_042_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER42;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER43 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER43 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_043/customer_043_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER43;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER44 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER44 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_044/customer_044_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER44;
```

COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER45 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER45 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_045/customer_045_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER45;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER46 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER46 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_046/customer_046_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER46;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER47 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER47 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_047/customer_047_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER47;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER48 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER48 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /fix8/flat_048/customer_048_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER48;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER49 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER49 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_049/customer_049_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER49;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER50 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER50 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_050/customer_050_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER50;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER51 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER51 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_051/customer_051_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER51;
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 /fix8/flat_005/customer_005_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER52 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER52 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_052/customer_052_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER52;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER53 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER53 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_053/customer_053_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER53;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER54 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER54 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_054/customer_054_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO CUSTOMER54;
COMMIT WORK;
CONNECT RESET;

LOAD CUSTOMER55 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER55 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_055/customer_055_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER55;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER56 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER56 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_056/customer_056_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER56;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER57 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER57 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_057/customer_057_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER57;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER58 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER58 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_058/customer_058_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER58;
```

```
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER59 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER59 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_059/customer_059_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER59;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER60 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER60 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_060/customer_060_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER60;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER61 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER61 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_061/customer_061_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER61;  
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 36000000 INSERT INTO CUSTOMER6;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER62 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER62 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_062/customer_062_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER62;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER63 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER63 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_063/customer_063_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER63;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER64 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER64 ACTIVATE NOT LOGGED  
INITIALLY;  
IMPORT FROM /flx8/flat_064/customer_064_1.dat OF DEL  
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-  
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50  
COMMITCOUNT 36000000 INSERT INTO CUSTOMER64;  
COMMIT WORK;  
CONNECT RESET;
```

LOAD CUSTOMER65 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER65 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_065/customer_065_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER65;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER66 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER66 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_066/customer_066_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER66;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER67 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER67 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_067/customer_067_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER67;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER68 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER68 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_068/customer_068_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER68;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER69 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER69 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_069/customer_069_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER69;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER70 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER70 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_070/customer_070_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER70;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER71 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER71 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_071/customer_071_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER71;
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 36000000 INSERT INTO CUSTOMER7;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER72 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER72 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_072/customer_072_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER72;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER73 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER73 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_073/customer_073_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER73;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER74 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER74 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_074/customer_074_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER74;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER75 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER75 ACTIVATE NOT LOGGED
INITIALLY;
```



```
IMPORT FROM /flx8/flat_075/customer_075_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER75;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER76 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER76 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_076/customer_076_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER76;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER77 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER77 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_077/customer_077_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER77;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER78 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER78 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_078/customer_078_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER78;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER79 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER79 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_079/customer_079_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER79;
COMMIT WORK;
CONNECT RESET;
```

LOAD CUSTOMER80 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER80 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_080/customer_080_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO CUSTOMER80;
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 36000000 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 36000000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT10 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT10 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_010/district_010_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT10;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT11 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT11 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_011/district_011_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT11;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT1 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT1 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_001/district_001_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT1;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT12 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT12 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_012/district_012_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT12;
```

COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT13 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_013/district_013_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT13;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT14 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_014/district_014_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT14;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT15 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_015/district_015_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT15;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT16 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT16 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_016/district_016_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT16;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT17 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_017/district_017_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT17;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT18 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_018/district_018_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT18;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT19 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_019/district_019_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT19;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT20 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_020/district_020_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT20;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT21 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_021/district_021_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT21;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT2 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_002/district_002_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT2;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT22 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_022/district_022_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT22;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT23 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT23 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_023/district_023_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT23;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT24 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT24 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_024/district_024_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT24;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT25 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT25 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_025/district_025_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT25;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT26 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT26 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_026/district_026_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT26;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT27 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT27 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_027/district_027_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT27;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT28 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT28 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_028/district_028_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT28;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT29 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT29 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_029/district_029_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT29;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT30 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT30 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_030/district_030_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT30;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT31 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT31 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_031/district_031_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT31;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT3 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT3 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_003/district_003_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT3;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT32 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT32 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_032/district_032_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT32;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT33 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT33 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_033/district_033_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT33;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT34 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT34 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_034/district_034_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT34;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT35 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT35 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_035/district_035_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT35;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT36 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT36 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_036/district_036_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT36;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT37 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT37 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_037/district_037_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT37;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT38 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT38 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_038/district_038_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT38;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT39 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT39 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_039/district_039_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT39;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT40 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT40 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_040/district_040_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT40;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT41 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT41 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_041/district_041_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT41;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT4 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT4 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_004/district_004_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT4;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT42 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT42 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_042/district_042_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT42;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT43 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT43 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_043/district_043_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT43;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT44 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT44 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_044/district_044_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT44;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT45 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT45 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_045/district_045_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT45;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT46 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT46 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_046/district_046_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT46;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT47 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT47 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_047/district_047_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT47;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT48 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT48 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_048/district_048_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT48;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT49 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT49 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_049/district_049_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT49;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT50 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT50 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_050/district_050_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT50;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT51 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT51 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_051/district_051_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT51;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT5 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT5 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_005/district_005_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT5;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT52 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT52 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_052/district_052_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT52;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT53 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT53 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_053/district_053_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT53;
```

COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT54 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT54 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_054/district_054_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT54;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT55 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT55 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_055/district_055_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT55;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT56 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT56 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_056/district_056_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT56;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT57 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT57 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_057/district_057_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT57;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT58 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT58 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_058/district_058_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT58;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT59 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT59 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_059/district_059_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT59;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT60 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT60 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_060/district_060_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT60;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT61 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT61 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_061/district_061_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT61;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT6 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_006/district_006_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT6;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT62 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT62 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_062/district_062_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT62;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT63 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT63 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_063/district_063_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 12000 INSERT INTO DISTRICT63;
COMMIT WORK;
CONNECT RESET;

LOAD DISTRICT64 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT64 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_064/district_064_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT64;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT65 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT65 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_065/district_065_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT65;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT66 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT66 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_066/district_066_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT66;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT67 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT67 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_067/district_067_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT67;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT68 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT68 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_068/district_068_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT68;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT69 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT69 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_069/district_069_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT69;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT70 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT70 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_070/district_070_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT70;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT71 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT71 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_071/district_071_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT71;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT7 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT7 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_007/district_007_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT7;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT72 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT72 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_072/district_072_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT72;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT73 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT73 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_073/district_073_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT73;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT74 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT74 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_074/district_074_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT74;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT75 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT75 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_075/district_075_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT75;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT76 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT76 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_076/district_076_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT76;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT77 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT77 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_077/district_077_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT77;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT78 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT78 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_078/district_078_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT78;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT79 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT79 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_079/district_079_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT79;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT80 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT80 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_080/district_080_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT80;
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT8 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT8 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_008/district_008_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 INSERT INTO DISTRICT8;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD DISTRICT9 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE DISTRICT9 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_009/district_009_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000 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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;

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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
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 1 ;
connect reset;

LOAD_HISTORY33.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_033/history_033_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY33 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD_HISTORY34.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_034/history_034_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY34 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD_HISTORY35.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_035/history_035_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY35 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD_HISTORY36.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_036/history_036_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY36 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD_HISTORY37.ddl

connect to TPCC in share mode;

LOAD FROM /flx8/flat_037/history_037_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY37 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD_HISTORY38.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_038/history_038_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY38 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD_HISTORY39.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_039/history_039_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY39 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
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 1 ;
connect reset;

LOAD_HISTORY40.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_040/history_040_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY40 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD_HISTORY41.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_041/history_041_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE

REPLACE INTO HISTORY41 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY42.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_042/history_042_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY42 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY43.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_043/history_043_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY43 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY44.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_044/history_044_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY44 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY45.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_045/history_045_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY45 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY46.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_046/history_046_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY46 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;

connect reset;

LOAD HISTORY47.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_047/history_047_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY47 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY48.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_048/history_048_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY48 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY49.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_049/history_049_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY49 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
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 1 ;
connect reset;

LOAD HISTORY50.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_050/history_050_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY50 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY51.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_051/history_051_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY51 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY52.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_052/history_052_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY52 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY53.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_053/history_053_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY53 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY54.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_054/history_054_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY54 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY55.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_055/history_055_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY55 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY56.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_056/history_056_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY56 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY57.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_057/history_057_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY57 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY58.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_058/history_058_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY58 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY59.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_059/history_059_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY59 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
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 1 ;
connect reset;

LOAD HISTORY60.ddl

connect to TPCC in share mode;

LOAD FROM /flx8/flat_060/history_060_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY60 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY61.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_061/history_061_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY61 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY62.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_062/history_062_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY62 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY63.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_063/history_063_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY63 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY64.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_064/history_064_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY64 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY65.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_065/history_065_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE

REPLACE INTO HISTORY65 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY66.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_066/history_066_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY66 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY67.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_067/history_067_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY67 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY68.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_068/history_068_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY68 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY69.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_069/history_069_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY69 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
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 1 ;

connect reset;

LOAD HISTORY70.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_070/history_070_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY70 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY71.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_071/history_071_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY71 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY72.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_072/history_072_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY72 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY73.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_073/history_073_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY73 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY74.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_074/history_074_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY74 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY75.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_075/history_075_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY75 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY76.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_076/history_076_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY76 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY77.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_077/history_077_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY77 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY78.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_078/history_078_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY78 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
connect reset;

LOAD HISTORY79.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_079/history_079_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY79 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
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 1 ;
connect reset;

LOAD HISTORY80.ddl

connect to TPCC in share mode;
LOAD FROM /flx8/flat_080/history_080_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY80 NONRECOVERABLE DATA
BUFFER 5000 CPU_PARALLELISM 1 ;
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 1 ;
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 1 ;
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;

```
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA10 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_010/neworder_010_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA10;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA11 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA11 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_011/neworder_011_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA11;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA1 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA1 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_001/neworder_001_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA1;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA12 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA12 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_012/neworder_012_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA12;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA13 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA13 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_013/neworder_013_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA13;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA14 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA14 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_014/neworder_014_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA14;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA15 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA15 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_015/neworder_015_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA15;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA16 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA16 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_016/neworder_016_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA16;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA17 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA17 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_017/neworder_017_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA17;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA18 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA18 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_018/neworder_018_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA18;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA19 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA19 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_019/neworder_019_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA19;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA20 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA20 ACTIVATE NOT
LOGGED INITIALLY;
```

```
IMPORT FROM /flx8/flat_020/neworder_020_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA20;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA21 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA21 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_021/neworder_021_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA21;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA2 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA2 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_002/neworder_002_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA2;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA22 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA22 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_022/neworder_022_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA22;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA23 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA23 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_023/neworder_023_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA23;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA24 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA24 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_024/neworder_024_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA24;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA25 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA25 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_025/neworder_025_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA25;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA26 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA26 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_026/neworder_026_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA26;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA27 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA27 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_027/neworder_027_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA27;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA28 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA28 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_028/neworder_028_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA28;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA29 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA29 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_029/neworder_029_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA29;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA30 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA30 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_030/neworder_030_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA30;
```

COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA31 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA31 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_031/neworder_031_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA31;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA3 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA3 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /fix8/flat_003/neworder_003_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA3;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA32 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA32 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_032/neworder_032_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA32;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA33 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA33 ACTIVATE NOT
LOGGED INITIALLY;

IMPORT FROM /fix8/flat_033/neworder_033_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA33;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA34 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA34 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_034/neworder_034_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA34;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA35 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA35 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_035/neworder_035_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA35;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA36 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA36 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_036/neworder_036_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA36;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA37 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA37 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_037/neworder_037_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA37;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA38 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA38 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_038/neworder_038_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA38;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA39 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA39 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_039/neworder_039_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA39;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA40 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA40 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /fix8/flat_040/neworder_040_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA40;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA41 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA41 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_041/neworder_041_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA41;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA4 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA4 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_004/neworder_004_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA4;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA42 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA42 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_042/neworder_042_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA42;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA43 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA43 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_043/neworder_043_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA43;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA44 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA44 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_044/neworder_044_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA44;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA45 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA45 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_045/neworder_045_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA45;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA46 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA46 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_046/neworder_046_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA46;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA47 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA47 ACTIVATE NOT
LOGGED INITIALLY;
```

```
IMPORT FROM /flx8/flat_047/neworder_047_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA47;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA48 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA48 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_048/neworder_048_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA48;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA49 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA49 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_049/neworder_049_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA49;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA50 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA50 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_050/neworder_050_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA50;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA51 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA51 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_051/neworder_051_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA51;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA5 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA5 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_005/neworder_005_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA5;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA52 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA52 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_052/neworder_052_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA52;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA53 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA53 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_053/neworder_053_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA53;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA54 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA54 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_054/neworder_054_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA54;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA55 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA55 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_055/neworder_055_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA55;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA56 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA56 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_056/neworder_056_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA56;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA57 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA57 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_057/neworder_057_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA57;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA58 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA58 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_058/neworder_058_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA58;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA59 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA59 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_059/neworder_059_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA59;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA60 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA60 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_060/neworder_060_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA60;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA61 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA61 ACTIVATE NOT
LOGGED INITIALLY;
```

```
IMPORT FROM /flx8/flat_061/neworder_061_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA61;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA6 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA6 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_006/neworder_006_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA6;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA62 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA62 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_062/neworder_062_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA62;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA63 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA63 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_063/neworder_063_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA63;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA64 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA64 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_064/neworder_064_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA64;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA65 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA65 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_065/neworder_065_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA65;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA66 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA66 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_066/neworder_066_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA66;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA67 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA67 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_067/neworder_067_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA67;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA68 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA68 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_068/neworder_068_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA68;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA69 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA69 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_069/neworder_069_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA69;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA70 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA70 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_070/neworder_070_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA70;
COMMIT WORK;
CONNECT RESET;
```

LOAD NEW ORDERA71 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA71 ACTIVATE NOT
LOGGED INITIALLY;
IMPORT FROM /flx8/flat_071/neworder_071_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 20000 INSERT INTO NEW_ORDERA71;
```

COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA7 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_007/neworder_007_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA7;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA72 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA72 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_072/neworder_072_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA72;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA73 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA73 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_073/neworder_073_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA73;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA74 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA74 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /fix8/flat_074/neworder_074_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA74;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA75 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA75 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_075/neworder_075_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA75;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA76 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA76 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_076/neworder_076_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA76;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA77 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA77 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_077/neworder_077_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA77;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA78 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA78 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_078/neworder_078_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA78;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA79 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA79 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_079/neworder_079_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA79;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA80 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA80 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_080/neworder_080_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA80;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA8 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_008/neworder_008_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 20000 INSERT INTO NEW_ORDERA8;
COMMIT WORK;
CONNECT RESET;

LOAD NEW ORDERA9 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDERA9 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_009/neworder_009_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
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 /flx8/flat_010/orderline_010_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_011/orderline_011_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_001/orderline_001_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_012/orderline_012_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_013/orderline_013_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_014/orderline_014_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_015/orderline_015_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_016/orderline_016_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_017/orderline_017_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_018/orderline_018_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_019/orderline_019_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_020/orderline_020_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_021/orderline_021_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_002/orderline_002_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_022/orderline_022_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_023/orderline_023_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_024/orderline_024_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_025/orderline_025_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_026/orderline_026_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_027/orderline_027_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_028/orderline_028_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_029/orderline_029_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_030/orderline_030_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_031/orderline_031_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_003/orderline_003_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 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 /flx8/flat_032/orderline_032_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE33 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_033/orderline_033_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE34 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_034/orderline_034_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE35 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_035/orderline_035_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE36 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_036/orderline_036_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE37 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_037/orderline_037_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE38 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_038/orderline_038_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE39 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE39 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_039/orderline_039_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE39;
```

COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE40 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE40 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_040/orderline_040_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE40;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE41 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE41 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_041/orderline_041_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE41;
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 /flx8/flat_004/orderline_004_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE42 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE42 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_042/orderline_042_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE42;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE43 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE43 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_043/orderline_043_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE43;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE44 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE44 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_044/orderline_044_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE44;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE45 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE45 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_045/orderline_045_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE45;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE46 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE46 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_046/orderline_046_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE46;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE47 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE47 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_047/orderline_047_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE47;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE48 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE48 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_048/orderline_048_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE48;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE49 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE49 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_049/orderline_049_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE49;
COMMIT WORK;
CONNECT RESET;

LOAD ORDER LINE50 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE50 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_050/orderline_050_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE50;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE51 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE51 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_051/orderline_051_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE51;
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 /flx8/flat_005/orderline_005_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE52 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE52 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_052/orderline_052_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE52;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE53 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE53 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_053/orderline_053_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE53;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE54 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE54 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_054/orderline_054_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE54;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE55 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE55 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_055/orderline_055_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE55;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE56 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE56 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_056/orderline_056_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE56;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE57 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE57 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_057/orderline_057_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE57;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE58 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE58 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_058/orderline_058_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE58;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE59 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE59 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_059/orderline_059_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE59;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE60 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE60 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_060/orderline_060_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE60;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE61 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE61 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_061/orderline_061_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE61;
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 /flx8/flat_006/orderline_006_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE62 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE62 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_062/orderline_062_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE62;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE63 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE63 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_063/orderline_063_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE63;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE64 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE64 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_064/orderline_064_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE64;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE65 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE65 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_065/orderline_065_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE65;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE66 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE66 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_066/orderline_066_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE66;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE67 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE67 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_067/orderline_067_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE67;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE68 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE68 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_068/orderline_068_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE68;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE69 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE69 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_069/orderline_069_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE69;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE70 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE70 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_070/orderline_070_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE70;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE71 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE71 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_071/orderline_071_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE71;
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 /flx8/flat_007/orderline_007_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE72 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE72 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_072/orderline_072_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE72;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE73 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE73 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_073/orderline_073_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE73;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE74 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE74 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_074/orderline_074_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE74;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE75 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE75 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_075/orderline_075_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE75;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE76 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE76 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_076/orderline_076_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE76;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE77 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE77 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_077/orderline_077_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE77;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE78 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE78 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_078/orderline_078_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE78;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE79 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE79 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_079/orderline_079_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE79;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDER LINE80 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE80 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_080/orderline_080_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 396000000 INSERT INTO ORDER_LINE80;
```

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 /fix8/flat_008/orderline_008_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 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 /fix8/flat_009/orderline_009_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 396000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS10 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_010/orders_010_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS10;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS11 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS11 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /fix8/flat_011/orders_011_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS11;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS1 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_001/orders_001_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS1;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS12 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_012/orders_012_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS12;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS13 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_013/orders_013_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS13;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS14 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_014/orders_014_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS14;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS15 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_015/orders_015_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS15;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS16 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_016/orders_016_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS16;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS17 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_017/orders_017_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 360000000 INSERT INTO ORDERS17;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS18 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS18 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_018/orders_018_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS18;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS19 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS19 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_019/orders_019_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS19;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS20 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS20 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_020/orders_020_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS20;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS21 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS21 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_021/orders_021_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS21;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS2 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS2 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_002/orders_002_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS2;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS22 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS22 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_022/orders_022_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS22;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS23 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS23 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_023/orders_023_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS23;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS24 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS24 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_024/orders_024_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS24;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS25 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS25 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_025/orders_025_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS25;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS26 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS26 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_026/orders_026_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS26;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS27 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS27 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_027/orders_027_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS27;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS28 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS28 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_028/orders_028_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS28;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS29 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS29 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_029/orders_029_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS29;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS30 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS30 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_030/orders_030_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS30;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS31 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS31 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_031/orders_031_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS31;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS3 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS3 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_003/orders_003_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS3;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS32 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS32 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_032/orders_032_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS32;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS33 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS33 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_033/orders_033_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS33;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS34 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS34 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_034/orders_034_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS34;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS35 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS35 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_035/orders_035_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS35;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS36 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS36 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_036/orders_036_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS36;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS37 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS37 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_037/orders_037_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS37;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS38 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS38 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_038/orders_038_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS38;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS39 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS39 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_039/orders_039_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS39;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS40 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS40 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_040/orders_040_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS40;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS41 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS41 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_041/orders_041_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS41;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS4 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS4 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_004/orders_004_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS4;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS42 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS42 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_042/orders_042_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS42;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS43 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS43 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_043/orders_043_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS43;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS44 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS44 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_044/orders_044_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS44;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS45 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS45 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_045/orders_045_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS45;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS46 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS46 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_046/orders_046_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS46;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS47 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS47 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_047/orders_047_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS47;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS48 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS48 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_048/orders_048_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS48;
```

COMMIT WORK;
CONNECT RESET;

LOAD ORDERS49 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS49 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_049/orders_049_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS49;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS50 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS50 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_050/orders_050_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS50;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS51 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS51 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_051/orders_051_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS51;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS5 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS5 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /fix8/flat_005/orders_005_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS5;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS52 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS52 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_052/orders_052_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS52;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS53 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS53 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_053/orders_053_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS53;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS54 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS54 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_054/orders_054_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS54;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS55 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS55 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_055/orders_055_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS55;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS56 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS56 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_056/orders_056_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS56;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS57 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS57 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_057/orders_057_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS57;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS58 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS58 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /fix8/flat_058/orders_058_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS58;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS59 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS59 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_059/orders_059_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS59;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS60 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS60 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_060/orders_060_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS60;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS61 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS61 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_061/orders_061_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS61;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS6 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_006/orders_006_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS6;

COMMIT WORK;
CONNECT RESET;

LOAD ORDERS62 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS62 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_062/orders_062_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS62;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS63 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS63 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_063/orders_063_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS63;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS64 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS64 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_064/orders_064_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS64;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS65 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS65 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_065/orders_065_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS65;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS66 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS66 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_066/orders_066_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS66;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS67 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS67 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_067/orders_067_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS67;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS68 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS68 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_068/orders_068_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 36000000 INSERT INTO ORDERS68;
COMMIT WORK;
CONNECT RESET;

LOAD ORDERS69 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS69 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_069/orders_069_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS69;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS70 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS70 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_070/orders_070_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS70;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS71 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS71 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_071/orders_071_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS71;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS7 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS7 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_007/orders_007_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS7;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS72 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS72 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_072/orders_072_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS72;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS73 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS73 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_073/orders_073_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS73;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS74 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS74 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_074/orders_074_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS74;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS75 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS75 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_075/orders_075_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS75;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS76 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS76 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_076/orders_076_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS76;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS77 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS77 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_077/orders_077_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS77;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS78 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS78 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_078/orders_078_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS78;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS79 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS79 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_079/orders_079_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS79;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS80 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS80 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_080/orders_080_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS80;
COMMIT WORK;
CONNECT RESET;
```

LOAD ORDERS8 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS8 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_008/orders_008_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 36000000 INSERT INTO ORDERS8;
COMMIT WORK;
CONNECT RESET;
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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 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 120000000 INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK33 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_033/stock_033_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 120000000 INSERT INTO STOCK33;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK34 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK34 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_034/stock_034_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 120000000 INSERT INTO STOCK34;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK35 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_035/stock_035_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 120000000 INSERT INTO STOCK35;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK36 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_036/stock_036_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 120000000 INSERT INTO STOCK36;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK37 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_037/stock_037_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 120000000 INSERT INTO STOCK37;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK38 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK38 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_038/stock_038_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK38;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK39 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK39 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_039/stock_039_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK39;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK40 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK40 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_040/stock_040_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK40;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK41 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK41 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_041/stock_041_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK41;
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 120000000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK42 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK42 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_042/stock_042_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK42;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK43 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK43 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_043/stock_043_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK43;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK44 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK44 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_044/stock_044_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK44;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK45 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK45 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_045/stock_045_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK45;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK46 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK46 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_046/stock_046_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK46;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK47 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK47 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_047/stock_047_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK47;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK48 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK48 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_048/stock_048_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK48;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK49 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK49 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_049/stock_049_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK49;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK50 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK50 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_050/stock_050_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK50;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK51 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK51 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_051/stock_051_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK51;
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 120000000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK52 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK52 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_052/stock_052_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK52;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK53 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK53 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_053/stock_053_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK53;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK54 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK54 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_054/stock_054_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK54;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK55 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK55 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_055/stock_055_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK55;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK56 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK56 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_056/stock_056_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK56;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK57 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK57 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_057/stock_057_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK57;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK58 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK58 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_058/stock_058_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK58;
```

COMMIT WORK;
CONNECT RESET;

LOAD STOCK59 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK59 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_059/stock_059_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK59;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK60 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK60 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_060/stock_060_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK60;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK61 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK61 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_061/stock_061_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK61;
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 120000000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK62 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK62 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_062/stock_062_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK62;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK63 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK63 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_063/stock_063_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK63;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK64 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK64 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_064/stock_064_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK64;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK65 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK65 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_065/stock_065_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK65;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK66 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK66 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_066/stock_066_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK66;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK67 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK67 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_067/stock_067_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK67;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK68 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK68 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_068/stock_068_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK68;
COMMIT WORK;
CONNECT RESET;

LOAD STOCK69 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK69 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_069/stock_069_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK69;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK70 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK70 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_070/stock_070_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK70;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK71 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK71 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_071/stock_071_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK71;
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 120000000 INSERT INTO STOCK7;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK72 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK72 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_072/stock_072_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK72;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK73 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK73 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_073/stock_073_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK73;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK74 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK74 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_074/stock_074_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK74;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK75 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK75 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_075/stock_075_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK75;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK76 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK76 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_076/stock_076_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK76;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK77 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK77 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_077/stock_077_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK77;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK78 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK78 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_078/stock_078_1.dat OF DEL
MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK78;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK79 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK79 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_079/stock_079_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK79;
COMMIT WORK;
CONNECT RESET;
```

LOAD STOCK80 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK80 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_080/stock_080_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 120000000 INSERT INTO STOCK80;
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 120000000 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 120000000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE10 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE10 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_010/warehouse_010_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE10;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE11 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE11 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_011/warehouse_011_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE11;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE1 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE1 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_001/warehouse_001_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE1;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE12 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE12 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_012/warehouse_012_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE12;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE13 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE13 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_013/warehouse_013_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE13;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE14 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE14 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_014/warehouse_014_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE14;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE15 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE15 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_015/warehouse_015_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE15;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE16 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE16 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_016/warehouse_016_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE16;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE17 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE17 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_017/warehouse_017_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE17;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE18 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE18 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_018/warehouse_018_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE18;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE19 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE19 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_019/warehouse_019_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE19;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE20 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE20 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_020/warehouse_020_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE20;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE21 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE21 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_021/warehouse_021_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE21;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE2 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE2 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_002/warehouse_002_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE2;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE22 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE22 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_022/warehouse_022_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE22;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE23 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE23 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_023/warehouse_023_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE23;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE24 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE24 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_024/warehouse_024_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE24;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE25 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE25 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_025/warehouse_025_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE25;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE26 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE26 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_026/warehouse_026_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE26;
```

COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE27 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_027/warehouse_027_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE27;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE28 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_028/warehouse_028_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE28;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE29 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_029/warehouse_029_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE29;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE30 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE30 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_030/warehouse_030_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE30;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE31 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_031/warehouse_031_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE31;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE3 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_003/warehouse_003_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE3;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE32 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_032/warehouse_032_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE32;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE33 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_033/warehouse_033_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE33;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE34 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_034/warehouse_034_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE34;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE35 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_035/warehouse_035_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE35;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE36 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_036/warehouse_036_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE36;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE37 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_037/warehouse_037_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE37;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE38 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_038/warehouse_038_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE38;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE39 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_039/warehouse_039_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE39;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE40 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE40 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_040/warehouse_040_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE40;

COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE41 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE41 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_041/warehouse_041_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE41;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE4 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_004/warehouse_004_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE4;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE42 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE42 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_042/warehouse_042_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE42;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE43 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE43 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_043/warehouse_043_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE43;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE44 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE44 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_044/warehouse_044_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE44;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE45 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE45 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_045/warehouse_045_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE45;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE46 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE46 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_046/warehouse_046_1.dat OF DEL MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE46;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE47 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE47 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_047/warehouse_047_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE47;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE48 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE48 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_048/warehouse_048_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE48;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE49 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE49 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_049/warehouse_049_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE49;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE50 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE50 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_050/warehouse_050_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE50;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE51 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE51 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_051/warehouse_051_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE51;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE5 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE5 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_005/warehouse_005_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE5;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE52 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE52 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_052/warehouse_052_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE52;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE53 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE53 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_053/warehouse_053_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE53;
```

```
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE54 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE54 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_054/warehouse_054_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE54;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE55 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE55 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_055/warehouse_055_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE55;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE56 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE56 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_056/warehouse_056_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE56;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE57 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE57 ACTIVATE NOT LOGGED
INITIALLY;
```

```
IMPORT FROM /flx8/flat_057/warehouse_057_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE57;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE58 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE58 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_058/warehouse_058_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE58;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE59 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE59 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_059/warehouse_059_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE59;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE60 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE60 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_060/warehouse_060_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE60;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE61 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE61 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_061/warehouse_061_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE61;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE6 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE6 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_006/warehouse_006_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE6;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE62 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE62 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_062/warehouse_062_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE62;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE63 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE63 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_063/warehouse_063_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE63;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE64 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE64 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_064/warehouse_064_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE64;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE65 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE65 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_065/warehouse_065_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE65;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE66 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE66 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_066/warehouse_066_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE66;
COMMIT WORK;
CONNECT RESET;
```

LOAD WAREHOUSE67 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE67 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flx8/flat_067/warehouse_067_1.dat OF DEL
MODIFIED BY COLDEL|TIMESTAMPFORMAT="YYYY-
MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50
COMMITCOUNT 1200 INSERT INTO WAREHOUSE67;
```

COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE68 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE68 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_068/warehouse_068_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE68;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE69 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE69 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_069/warehouse_069_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE69;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE70 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE70 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_070/warehouse_070_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE70;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE71 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE71 ACTIVATE NOT LOGGED INITIALLY;

IMPORT FROM /flx8/flat_071/warehouse_071_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE71;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE7 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_007/warehouse_007_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE7;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE72 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE72 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_072/warehouse_072_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE72;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE73 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE73 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_073/warehouse_073_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE73;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE74 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE74 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_074/warehouse_074_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE74;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE75 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE75 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_075/warehouse_075_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE75;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE76 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE76 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_076/warehouse_076_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE76;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE77 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE77 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_077/warehouse_077_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE77;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE78 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE78 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_078/warehouse_078_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE78;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE79 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE79 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_079/warehouse_079_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE79;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE80 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE80 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_080/warehouse_080_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE80;
COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE8 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_008/warehouse_008_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 INSERT INTO WAREHOUSE8;

COMMIT WORK;
CONNECT RESET;

LOAD WAREHOUSE9 1.ddl

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE WAREHOUSE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flx8/flat_009/warehouse_009_1.dat OF DEL MODIFIED BY COLDEL| TIMESTAMPFORMAT="YYYY-MM-DD HH:MM:SS" KEEPBLANKS COMPOUND=50 COMMITCOUNT 1200 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_CUSTOMER33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER33 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER34 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER35 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER36 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER37 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER38 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER39 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_CUSTOMER40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER40 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER41 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER42 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER43 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER44 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER45 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER46 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER47 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER48 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER49 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_CUSTOMER50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER50 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER51 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER52 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER53 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER54 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER55 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER56 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER57 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER58 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER59 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_CUSTOMER60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER60 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER61 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER62 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER63 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER64 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER65 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER66 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER67 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER68 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER69 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_CUSTOMER70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER70 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER71 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER72 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER73 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER74 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER75 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER76 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER77 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER78 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER79 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER7 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST_CUSTOMER80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER80 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_DISTRICT33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT33 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT34 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT35 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT36 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT37 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT38 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT39 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_DISTRICT40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT41 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT42 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT43 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT44 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT45 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT46 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT47 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT48 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT49 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_DISTRICT50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT50 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT51 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT52 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT53 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT54 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT55 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT56 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT57 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT58 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT59 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_DISTRICT60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT60 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT61 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT62 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT63 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT64 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT65 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT66 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT67 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT68 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT69 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_DISTRICT70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT70 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT71 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT72 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT73 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT74 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT75 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT76 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT77 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT78 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT79 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_DISTRICT80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT80 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 HISTORY33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY33 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY34 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY35 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY36 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY37 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY38 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY39 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 HISTORY40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY40 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY41 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY42 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY43 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY44 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY45 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY46 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY47 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY48 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY49 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 HISTORY50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY50 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY51 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY52 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY53 AND INDEXES
ALL;
COMMIT WORK;
connect reset;

RNST HISTORY54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY54 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY55 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY56 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY57 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY58 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY59 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 HISTORY60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY60 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY61 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY62 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY63 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY64 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY65 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY66 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY67 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY68 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY69 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 HISTORY70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY70 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY71 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY72 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY73 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY74 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY75 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY76 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY77 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY78 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST HISTORY79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY79 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 HISTORY80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY80 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_ORDERA33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA33 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA34 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA35 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA36 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA37 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA38 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA39 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_ORDERA40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA40 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA41 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA42 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA43 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA44 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA45 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA46 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA47 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA48 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA49 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_ORDERA50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA50 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA51 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA52 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA53 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA54 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA55 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA56 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA57 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA58 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA59 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_ORDERA60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA60 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA61 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA62 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA63 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA64 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA65 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA66 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA67 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA68 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA69 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_ORDERA70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA70 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA71 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA72 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA73 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA74 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA75 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA76 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA77 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA78 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERA79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA79 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_ORDERA80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA80 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_ORDERB33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB33 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB34 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB35 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB36 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB37 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB38 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB39 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_ORDERB40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB40 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB41 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB42 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB43 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB44 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB45 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB46 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB47 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB48 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB49 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_ORDERB50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB50 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB51 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB52 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB53 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB54 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB55 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB56 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB57 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB58 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB59 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_ORDERB60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB60 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB61 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB62 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB63 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB64 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB65 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB66 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB67 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB68 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB69 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_ORDERB70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB70 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB71 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB72 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB73 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB74 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB75 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB76 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB77 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB78 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDERB79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB79 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_ORDERB80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB80 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 LINE33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE33 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE34 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE35 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE36 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE37 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE38 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE39 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 LINE40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE40 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE41 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE42 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE43 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE44 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE45 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE46 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE47 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE48 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE49 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 LINE50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE50 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE51 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE52 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE53 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE54 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE55 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE56 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE57 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE58 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE59 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 LINE60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE60 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE61 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE62 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE63 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE64 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE65 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE66 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE67 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE68 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE69 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 LINE70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE70 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE71 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE72 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE73 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE74 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE75 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE76 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE77 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE78 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST ORDER LINE79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE79 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 LINE80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDER_LINE80 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_ORDERS33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS33 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS34 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS35 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS36 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS37 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS38 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS39 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_ORDERS40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS41 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS42 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS43 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS44 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS45 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS46 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS47 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS48 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS49 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_ORDERS50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS50 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS51 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS52 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS53 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS54 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS55 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS56 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS57 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS58 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS59 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_ORDERS60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS60 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS61 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS62 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS63 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS64 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS65 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS66 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS67 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS68 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS69 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_ORDERS70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS70 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS71 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS72 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS73 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS74 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS75 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS76 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS77 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS78 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_ORDERS79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS79 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_ORDERS80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ORDERS80 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 STOCK33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK33 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK34 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK35.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE tpcc.STOCK35 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK36 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK37 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK38 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK39 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 STOCK40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK40 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK41 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK42 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK43 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK44 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK45 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK46 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK47.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE tpcc.STOCK47 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK48 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK49 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 STOCK50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK50 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK51 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK52 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK53 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK54 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK55 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK56 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK57 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK58 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK59.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE tpcc.STOCK59 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 STOCK60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK60 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK61 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK62 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK63 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK64 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK65 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK66 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK67 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK68 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK69 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 STOCK70.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE tpcc.STOCK70 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK71 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK72 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK73 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK74 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK75 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK76 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK77 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK78 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST STOCK79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK79 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 STOCK80.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK80 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 WAREHOUSE33.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE33 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE34.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE34 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE35.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE35 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE36.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE36 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE37.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE37 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE38.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE38 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE39.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE39 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 WAREHOUSE40.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE40 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE41.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE41 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE42.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE42 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE43.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE43 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE44.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE44 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE45.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE45 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE46.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE46 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE47.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE47 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE48.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE48 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE49.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE49 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 WAREHOUSE50.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE50 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE51.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE51 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE52.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE52 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE53.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE53 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE54.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE54 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE55.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE55 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE56.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE56 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE57.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE57 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE58.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE58 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE59.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE59 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 WAREHOUSE60.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE60 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE61.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE61 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE62.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE62 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE63.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE63 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE64.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE64 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE65.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE65 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE66.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE66 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE67.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE67 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE68.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE68 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE69.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE69 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 WAREHOUSE70.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE70 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE71.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE71 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE72.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE72 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE73.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE73 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE74.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE74 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE75.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE75 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE76.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE76 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE77.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE77 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE78.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE78 AND
INDEXES ALL;
COMMIT WORK;
connect reset;

RNST WAREHOUSE79.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE79 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 WAREHOUSE80.ddl


```
connect to TPCC in share mode;  
RUNSTATS ON TABLE tpcc.WAREHOUSE80 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;
```



```

alter tablespace ts_newordB_020 bufferpool NEW2;
alter tablespace ts_newordB_021 bufferpool NEW3;
alter tablespace ts_newordB_022 bufferpool NEW3;
alter tablespace ts_newordB_023 bufferpool NEW3;
alter tablespace ts_newordB_024 bufferpool NEW3;
alter tablespace ts_newordB_025 bufferpool NEW3;
alter tablespace ts_newordB_026 bufferpool NEW3;
alter tablespace ts_newordB_027 bufferpool NEW3;
alter tablespace ts_newordB_028 bufferpool NEW3;
alter tablespace ts_newordB_029 bufferpool NEW3;
alter tablespace ts_newordB_030 bufferpool NEW3;
alter tablespace ts_newordB_031 bufferpool NEW4;
alter tablespace ts_newordB_032 bufferpool NEW4;
alter tablespace ts_newordB_033 bufferpool NEW4;
alter tablespace ts_newordB_034 bufferpool NEW4;
alter tablespace ts_newordB_035 bufferpool NEW4;
alter tablespace ts_newordB_036 bufferpool NEW4;
alter tablespace ts_newordB_037 bufferpool NEW4;
alter tablespace ts_newordB_038 bufferpool NEW4;
alter tablespace ts_newordB_039 bufferpool NEW4;
alter tablespace ts_newordB_040 bufferpool NEW4;
alter tablespace ts_newordB_041 bufferpool NEW5;
alter tablespace ts_newordB_042 bufferpool NEW5;
alter tablespace ts_newordB_043 bufferpool NEW5;
alter tablespace ts_newordB_044 bufferpool NEW5;
alter tablespace ts_newordB_045 bufferpool NEW5;
alter tablespace ts_newordB_046 bufferpool NEW5;
alter tablespace ts_newordB_047 bufferpool NEW5;
alter tablespace ts_newordB_048 bufferpool NEW5;
alter tablespace ts_newordB_049 bufferpool NEW5;
alter tablespace ts_newordB_050 bufferpool NEW5;
alter tablespace ts_newordB_051 bufferpool NEW6;
alter tablespace ts_newordB_052 bufferpool NEW6;
alter tablespace ts_newordB_053 bufferpool NEW6;
alter tablespace ts_newordB_054 bufferpool NEW6;
alter tablespace ts_newordB_055 bufferpool NEW6;
alter tablespace ts_newordB_056 bufferpool NEW6;
alter tablespace ts_newordB_057 bufferpool NEW6;
alter tablespace ts_newordB_058 bufferpool NEW6;
alter tablespace ts_newordB_059 bufferpool NEW6;
alter tablespace ts_newordB_060 bufferpool NEW6;
alter tablespace ts_newordB_061 bufferpool NEW7;
alter tablespace ts_newordB_062 bufferpool NEW7;
alter tablespace ts_newordB_063 bufferpool NEW7;
alter tablespace ts_newordB_064 bufferpool NEW7;
alter tablespace ts_newordB_065 bufferpool NEW7;
alter tablespace ts_newordB_066 bufferpool NEW7;
alter tablespace ts_newordB_067 bufferpool NEW7;
alter tablespace ts_newordB_068 bufferpool NEW7;
alter tablespace ts_newordB_069 bufferpool NEW7;
alter tablespace ts_newordB_070 bufferpool NEW7;
alter tablespace ts_newordB_071 bufferpool NEW8;

```

```

alter tablespace ts_newordB_072 bufferpool NEW8;
alter tablespace ts_newordB_073 bufferpool NEW8;
alter tablespace ts_newordB_074 bufferpool NEW8;
alter tablespace ts_newordB_075 bufferpool NEW8;
alter tablespace ts_newordB_076 bufferpool NEW8;
alter tablespace ts_newordB_077 bufferpool NEW8;
alter tablespace ts_newordB_078 bufferpool NEW8;
alter tablespace ts_newordB_079 bufferpool NEW8;
alter tablespace ts_newordB_080 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 using codeset ISO8859-1
territory US 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' 400
)
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' 400

```

```

)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
)
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' 400
    )
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    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' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_033;
create regular tablespace ts_wh_033 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw33' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_034;
create regular tablespace ts_wh_034 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw34' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_035;
create regular tablespace ts_wh_035 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw35' 400

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_036;
create regular tablespace ts_wh_036 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw36' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_037;
create regular tablespace ts_wh_037 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw37' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_038;
create regular tablespace ts_wh_038 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw38' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_039;
create regular tablespace ts_wh_039 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw39' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_040;
create regular tablespace ts_wh_040 pagesize 4K
managed by database

```

```

using
(
    device '/dev/raw/raw40' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

drop tablespace ts_wh_041;
create regular tablespace ts_wh_041 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw41' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_042;
create regular tablespace ts_wh_042 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw42' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_043;
create regular tablespace ts_wh_043 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw43' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_044;
create regular tablespace ts_wh_044 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw44' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP

```

```

    prefetchsize 4096;
commit;
drop tablespace ts_wh_045;
create regular tablespace ts_wh_045 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw45' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_046;
create regular tablespace ts_wh_046 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw46' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_047;
create regular tablespace ts_wh_047 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw47' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_048;
create regular tablespace ts_wh_048 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw48' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_049;
create regular tablespace ts_wh_049 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw49' 400

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_050;
create regular tablespace ts_wh_050 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw50' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_051;
create regular tablespace ts_wh_051 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw51' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_052;
create regular tablespace ts_wh_052 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw52' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_053;
create regular tablespace ts_wh_053 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw53' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_054;
create regular tablespace ts_wh_054 pagesize 4K
managed by database

```

```

using
(
    device '/dev/raw/raw54' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_055;
create regular tablespace ts_wh_055 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw55' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_056;
create regular tablespace ts_wh_056 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw56' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_057;
create regular tablespace ts_wh_057 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw57' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_058;
create regular tablespace ts_wh_058 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw58' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_wh_059;
create regular tablespace ts_wh_059 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw59' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_060;
create regular tablespace ts_wh_060 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw60' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_061;
create regular tablespace ts_wh_061 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw61' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_062;
create regular tablespace ts_wh_062 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw62' 400
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_063;
create regular tablespace ts_wh_063 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw63' 400
)

```

```

    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_064;
create regular tablespace ts_wh_064 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw64' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_065;
create regular tablespace ts_wh_065 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw65' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_066;
create regular tablespace ts_wh_066 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw66' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_067;
create regular tablespace ts_wh_067 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw67' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_068;
create regular tablespace ts_wh_068 pagesize 4K
    managed by database
    using

```

```

    (
        device '/dev/raw/raw68' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_069;
create regular tablespace ts_wh_069 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw69' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_070;
create regular tablespace ts_wh_070 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw70' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_071;
create regular tablespace ts_wh_071 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw71' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_072;
create regular tablespace ts_wh_072 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw72' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_073;

```

```

create regular tablespace ts_wh_073 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw73' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_074;
create regular tablespace ts_wh_074 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw74' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_075;
create regular tablespace ts_wh_075 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw75' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_076;
create regular tablespace ts_wh_076 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw76' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_077;
create regular tablespace ts_wh_077 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw77' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP

```

```

        prefetchsize 4096;
commit;
drop tablespace ts_wh_078;
create regular tablespace ts_wh_078 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw78' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_079;
create regular tablespace ts_wh_079 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw79' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_wh_080;
create regular tablespace ts_wh_080 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw80' 400
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

drop tablespace ts_dis_001;
create regular tablespace ts_dis_001 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw81' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_002;
create regular tablespace ts_dis_002 pagesize 4K

```

```

    managed by database
    using
    (
        device '/dev/raw/raw82' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_003;
create regular tablespace ts_dis_003 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw83' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_004;
create regular tablespace ts_dis_004 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw84' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_005;
create regular tablespace ts_dis_005 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw85' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_006;
create regular tablespace ts_dis_006 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw86' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;

```

```

commit;
drop tablespace ts_dis_007;
create regular tablespace ts_dis_007 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw87' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_008;
create regular tablespace ts_dis_008 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw88' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_009;
create regular tablespace ts_dis_009 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw89' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_010;
create regular tablespace ts_dis_010 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw90' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_011;
create regular tablespace ts_dis_011 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw91' 900
    )

```

```

    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_012;
create regular tablespace ts_dis_012 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw92' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_013;
create regular tablespace ts_dis_013 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw93' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_014;
create regular tablespace ts_dis_014 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw94' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_015;
create regular tablespace ts_dis_015 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw95' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_016;
create regular tablespace ts_dis_016 pagesize 4K
    managed by database
    using

```

```

    (
        device '/dev/raw/raw96' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_017;
create regular tablespace ts_dis_017 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw97' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_018;
create regular tablespace ts_dis_018 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw98' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_019;
create regular tablespace ts_dis_019 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw99' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_020;
create regular tablespace ts_dis_020 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw100' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_021;

```

```

create regular tablespace ts_dis_021 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw101' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

drop tablespace ts_dis_022;
create regular tablespace ts_dis_022 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw102' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_023;
create regular tablespace ts_dis_023 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw103' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_024;
create regular tablespace ts_dis_024 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw104' 900
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_dis_025;
create regular tablespace ts_dis_025 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw105' 900
    )
    extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_026;
create regular tablespace ts_dis_026 pagesize 4K
managed by database
using
(
device '/dev/raw/raw106' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_027;
create regular tablespace ts_dis_027 pagesize 4K
managed by database
using
(
device '/dev/raw/raw107' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_028;
create regular tablespace ts_dis_028 pagesize 4K
managed by database
using
(
device '/dev/raw/raw108' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_029;
create regular tablespace ts_dis_029 pagesize 4K
managed by database
using
(
device '/dev/raw/raw109' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_030;
create regular tablespace ts_dis_030 pagesize 4K
managed by database
using
(

```

```

device '/dev/raw/raw110' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_031;
create regular tablespace ts_dis_031 pagesize 4K
managed by database
using
(
device '/dev/raw/raw111' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_032;
create regular tablespace ts_dis_032 pagesize 4K
managed by database
using
(
device '/dev/raw/raw112' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_033;
create regular tablespace ts_dis_033 pagesize 4K
managed by database
using
(
device '/dev/raw/raw113' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_034;
create regular tablespace ts_dis_034 pagesize 4K
managed by database
using
(
device '/dev/raw/raw114' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_035;
create regular tablespace ts_dis_035 pagesize 4K

```

```

managed by database
using
(
device '/dev/raw/raw115' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_036;
create regular tablespace ts_dis_036 pagesize 4K
managed by database
using
(
device '/dev/raw/raw116' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_037;
create regular tablespace ts_dis_037 pagesize 4K
managed by database
using
(
device '/dev/raw/raw117' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_038;
create regular tablespace ts_dis_038 pagesize 4K
managed by database
using
(
device '/dev/raw/raw118' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_dis_039;
create regular tablespace ts_dis_039 pagesize 4K
managed by database
using
(
device '/dev/raw/raw119' 900
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;

```



```

commit;
drop tablespace ts_dis_040;
create regular tablespace ts_dis_040 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw120' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;

drop tablespace ts_dis_041;
create regular tablespace ts_dis_041 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw121' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_042;
create regular tablespace ts_dis_042 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw122' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_043;
create regular tablespace ts_dis_043 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw123' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_044;
create regular tablespace ts_dis_044 pagesize 4K
  managed by database
  using
  (

```

```

    device '/dev/raw/raw124' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_045;
create regular tablespace ts_dis_045 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw125' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_046;
create regular tablespace ts_dis_046 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw126' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_047;
create regular tablespace ts_dis_047 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw127' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_048;
create regular tablespace ts_dis_048 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw128' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_049;
create regular tablespace ts_dis_049 pagesize 4K

```

```

  managed by database
  using
  (
    device '/dev/raw/raw129' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_050;
create regular tablespace ts_dis_050 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw130' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_051;
create regular tablespace ts_dis_051 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw131' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_052;
create regular tablespace ts_dis_052 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw132' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_053;
create regular tablespace ts_dis_053 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw133' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;

```

```

commit;
drop tablespace ts_dis_054;
create regular tablespace ts_dis_054 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw134' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_055;
create regular tablespace ts_dis_055 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw135' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_056;
create regular tablespace ts_dis_056 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw136' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_057;
create regular tablespace ts_dis_057 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw137' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_058;
create regular tablespace ts_dis_058 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw138' 900
  )

```

```

  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_059;
create regular tablespace ts_dis_059 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw139' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_060;
create regular tablespace ts_dis_060 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw140' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_061;
create regular tablespace ts_dis_061 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw141' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_062;
create regular tablespace ts_dis_062 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw142' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_063;
create regular tablespace ts_dis_063 pagesize 4K
  managed by database

```

```

  using
  (
    device '/dev/raw/raw143' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_064;
create regular tablespace ts_dis_064 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw144' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_065;
create regular tablespace ts_dis_065 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw145' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_066;
create regular tablespace ts_dis_066 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw146' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_067;
create regular tablespace ts_dis_067 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw147' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;

```

```

drop tablespace ts_dis_068;
create regular tablespace ts_dis_068 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw148' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_069;
create regular tablespace ts_dis_069 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw149' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_070;
create regular tablespace ts_dis_070 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw150' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_071;
create regular tablespace ts_dis_071 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw151' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_072;
create regular tablespace ts_dis_072 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw152' 900
  )
  extentsize 32

```

```

  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_073;
create regular tablespace ts_dis_073 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw153' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_074;
create regular tablespace ts_dis_074 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw154' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_075;
create regular tablespace ts_dis_075 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw155' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_076;
create regular tablespace ts_dis_076 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw156' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_077;
create regular tablespace ts_dis_077 pagesize 4K
  managed by database
  using
  (

```

```

    device '/dev/raw/raw157' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_078;
create regular tablespace ts_dis_078 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw158' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_079;
create regular tablespace ts_dis_079 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw159' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_dis_080;
create regular tablespace ts_dis_080 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw160' 900
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;

drop tablespace ts_item;
create regular tablespace ts_item pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw161' 100,
    device '/dev/raw/raw162' 100,

```

```

device '/dev/raw/raw163' 100,
device '/dev/raw/raw164' 100,
device '/dev/raw/raw165' 100,
device '/dev/raw/raw166' 100,
device '/dev/raw/raw167' 100,
device '/dev/raw/raw168' 100,
device '/dev/raw/raw169' 100,
device '/dev/raw/raw170' 100,
device '/dev/raw/raw171' 100,
device '/dev/raw/raw172' 100,
device '/dev/raw/raw173' 100,
device '/dev/raw/raw174' 100,
device '/dev/raw/raw175' 100,
device '/dev/raw/raw176' 100,
device '/dev/raw/raw177' 100,
device '/dev/raw/raw178' 100,
device '/dev/raw/raw179' 100,
device '/dev/raw/raw180' 100,
device '/dev/raw/raw181' 100,
device '/dev/raw/raw182' 100,
device '/dev/raw/raw183' 100,
device '/dev/raw/raw184' 100,
device '/dev/raw/raw185' 100,
device '/dev/raw/raw186' 100,
device '/dev/raw/raw187' 100,
device '/dev/raw/raw188' 100,
device '/dev/raw/raw189' 100,
device '/dev/raw/raw190' 100,
device '/dev/raw/raw191' 100,
device '/dev/raw/raw192' 100,
device '/dev/raw/raw193' 100,
device '/dev/raw/raw194' 100,
device '/dev/raw/raw195' 100,
device '/dev/raw/raw196' 100,
device '/dev/raw/raw197' 100,
device '/dev/raw/raw198' 100,
device '/dev/raw/raw199' 100,
device '/dev/raw/raw200' 100,
device '/dev/raw/raw201' 100,
device '/dev/raw/raw202' 100,
device '/dev/raw/raw203' 100,
device '/dev/raw/raw204' 100,
device '/dev/raw/raw205' 100,
device '/dev/raw/raw206' 100,
device '/dev/raw/raw207' 100,
device '/dev/raw/raw208' 100,
device '/dev/raw/raw209' 100,
device '/dev/raw/raw210' 100,
device '/dev/raw/raw211' 100,
device '/dev/raw/raw212' 100,
device '/dev/raw/raw213' 100,
device '/dev/raw/raw214' 100,

```

```

device '/dev/raw/raw215' 100,
device '/dev/raw/raw216' 100,
device '/dev/raw/raw217' 100,
device '/dev/raw/raw218' 100,
device '/dev/raw/raw219' 100,
device '/dev/raw/raw220' 100,
device '/dev/raw/raw221' 100,
device '/dev/raw/raw222' 100,
device '/dev/raw/raw223' 100,
device '/dev/raw/raw224' 100,
device '/dev/raw/raw225' 100,
device '/dev/raw/raw226' 100,
device '/dev/raw/raw227' 100,
device '/dev/raw/raw228' 100,
device '/dev/raw/raw229' 100,
device '/dev/raw/raw230' 100,
device '/dev/raw/raw231' 100,
device '/dev/raw/raw232' 100,
device '/dev/raw/raw233' 100,
device '/dev/raw/raw234' 100,
device '/dev/raw/raw235' 100,
device '/dev/raw/raw236' 100,
device '/dev/raw/raw237' 100,
device '/dev/raw/raw238' 100,
device '/dev/raw/raw239' 100,
device '/dev/raw/raw240' 100
)
extentsize 16
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;

drop tablespace ts_stock_001;
create regular tablespace ts_stock_001 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw241' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_002;
create regular tablespace ts_stock_002 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw242' 11000000
)
extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_003;
create regular tablespace ts_stock_003 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw243' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_004;
create regular tablespace ts_stock_004 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw244' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_005;
create regular tablespace ts_stock_005 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw245' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_006;
create regular tablespace ts_stock_006 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw246' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_007;
create regular tablespace ts_stock_007 pagesize 4K
managed by database
using
(

```

```

        device '/dev/raw/raw247' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_008;
create regular tablespace ts_stock_008 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw248' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_009;
create regular tablespace ts_stock_009 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw249' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_010;
create regular tablespace ts_stock_010 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw250' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_011;
create regular tablespace ts_stock_011 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw251' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_012;
create regular tablespace ts_stock_012 pagesize 4K

```

```

    managed by database
    using
    (
        device '/dev/raw/raw252' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_013;
create regular tablespace ts_stock_013 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw253' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_014;
create regular tablespace ts_stock_014 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw254' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_015;
create regular tablespace ts_stock_015 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw255' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_016;
create regular tablespace ts_stock_016 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw256' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;

```

```

commit;
drop tablespace ts_stock_017;
create regular tablespace ts_stock_017 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw257' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_018;
create regular tablespace ts_stock_018 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw258' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_019;
create regular tablespace ts_stock_019 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw259' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_020;
create regular tablespace ts_stock_020 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw260' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_021;
create regular tablespace ts_stock_021 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw261' 11000000
    )

```

```

    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

drop tablespace ts_stock_022;
create regular tablespace ts_stock_022 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw262' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_023;
create regular tablespace ts_stock_023 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw263' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_024;
create regular tablespace ts_stock_024 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw264' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_025;
create regular tablespace ts_stock_025 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw265' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_026;
create regular tablespace ts_stock_026 pagesize 4K
    managed by database

```

```

    using
    (
        device '/dev/raw/raw266' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_027;
create regular tablespace ts_stock_027 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw267' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_028;
create regular tablespace ts_stock_028 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw268' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_029;
create regular tablespace ts_stock_029 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw269' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_030;
create regular tablespace ts_stock_030 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw270' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_stock_031;
create regular tablespace ts_stock_031 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw271' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_032;
create regular tablespace ts_stock_032 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw272' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_033;
create regular tablespace ts_stock_033 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw273' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_034;
create regular tablespace ts_stock_034 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw274' 11000000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_035;
create regular tablespace ts_stock_035 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw275' 11000000
    )
    extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_036;
create regular tablespace ts_stock_036 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw276' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_037;
create regular tablespace ts_stock_037 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw277' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_038;
create regular tablespace ts_stock_038 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw278' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_039;
create regular tablespace ts_stock_039 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw279' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_040;
create regular tablespace ts_stock_040 pagesize 4K
managed by database
using
(

```

```

    device '/dev/raw/raw280' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_041;
create regular tablespace ts_stock_041 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw281' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_042;
create regular tablespace ts_stock_042 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw282' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_043;
create regular tablespace ts_stock_043 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw283' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_044;
create regular tablespace ts_stock_044 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw284' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

```

```

drop tablespace ts_stock_045;
create regular tablespace ts_stock_045 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw285' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_046;
create regular tablespace ts_stock_046 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw286' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_047;
create regular tablespace ts_stock_047 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw287' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_048;
create regular tablespace ts_stock_048 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw288' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_049;
create regular tablespace ts_stock_049 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw289' 11000000
)
extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_050;
create regular tablespace ts_stock_050 pagesize 4K
managed by database
using
(
device '/dev/raw/raw290' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_051;
create regular tablespace ts_stock_051 pagesize 4K
managed by database
using
(
device '/dev/raw/raw291' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_052;
create regular tablespace ts_stock_052 pagesize 4K
managed by database
using
(
device '/dev/raw/raw292' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_053;
create regular tablespace ts_stock_053 pagesize 4K
managed by database
using
(
device '/dev/raw/raw293' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_054;
create regular tablespace ts_stock_054 pagesize 4K
managed by database
using
(

```

```

device '/dev/raw/raw294' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_055;
create regular tablespace ts_stock_055 pagesize 4K
managed by database
using
(
device '/dev/raw/raw295' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_056;
create regular tablespace ts_stock_056 pagesize 4K
managed by database
using
(
device '/dev/raw/raw296' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_057;
create regular tablespace ts_stock_057 pagesize 4K
managed by database
using
(
device '/dev/raw/raw297' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_058;
create regular tablespace ts_stock_058 pagesize 4K
managed by database
using
(
device '/dev/raw/raw298' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_059;
create regular tablespace ts_stock_059 pagesize 4K

```

```

managed by database
using
(
device '/dev/raw/raw299' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_060;
create regular tablespace ts_stock_060 pagesize 4K
managed by database
using
(
device '/dev/raw/raw300' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_061;
create regular tablespace ts_stock_061 pagesize 4K
managed by database
using
(
device '/dev/raw/raw301' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_062;
create regular tablespace ts_stock_062 pagesize 4K
managed by database
using
(
device '/dev/raw/raw302' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_063;
create regular tablespace ts_stock_063 pagesize 4K
managed by database
using
(
device '/dev/raw/raw303' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP

```



```

        prefetchsize 4096;
commit;
drop tablespace ts_stock_064;
create regular tablespace ts_stock_064 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw304' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_065;
create regular tablespace ts_stock_065 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw305' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_066;
create regular tablespace ts_stock_066 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw306' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_067;
create regular tablespace ts_stock_067 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw307' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_068;
create regular tablespace ts_stock_068 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw308' 11000000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_stock_069;
create regular tablespace ts_stock_069 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw309' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_070;
create regular tablespace ts_stock_070 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw310' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_071;
create regular tablespace ts_stock_071 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw311' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_072;
create regular tablespace ts_stock_072 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw312' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_073;
create regular tablespace ts_stock_073 pagesize 4K
  managed by database

```

```

  using
  (
    device '/dev/raw/raw313' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_074;
create regular tablespace ts_stock_074 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw314' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_075;
create regular tablespace ts_stock_075 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw315' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_076;
create regular tablespace ts_stock_076 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw316' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_stock_077;
create regular tablespace ts_stock_077 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw317' 11000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;

```

```

drop tablespace ts_stock_078;
create regular tablespace ts_stock_078 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw318' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_079;
create regular tablespace ts_stock_079 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw319' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_stock_080;
create regular tablespace ts_stock_080 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw320' 11000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

drop tablespace ts_customer_001;
create regular tablespace ts_customer_001 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw321' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_002;
create regular tablespace ts_customer_002 pagesize 4K
managed by database
using
(

```

```

    device '/dev/raw/raw322' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_003;
create regular tablespace ts_customer_003 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw323' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_004;
create regular tablespace ts_customer_004 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw324' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_005;
create regular tablespace ts_customer_005 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw325' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_006;
create regular tablespace ts_customer_006 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw326' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_007;
create regular tablespace ts_customer_007 pagesize 4K

```

```

managed by database
using
(
    device '/dev/raw/raw327' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_008;
create regular tablespace ts_customer_008 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw328' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_009;
create regular tablespace ts_customer_009 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw329' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_010;
create regular tablespace ts_customer_010 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw330' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_011;
create regular tablespace ts_customer_011 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw331' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;

```

```

commit;
drop tablespace ts_customer_012;
create regular tablespace ts_customer_012 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw332' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_013;
create regular tablespace ts_customer_013 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw333' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_014;
create regular tablespace ts_customer_014 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw334' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_015;
create regular tablespace ts_customer_015 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw335' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_016;
create regular tablespace ts_customer_016 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw336' 8000000
  )

```

```

  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_017;
create regular tablespace ts_customer_017 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw337' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_018;
create regular tablespace ts_customer_018 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw338' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_019;
create regular tablespace ts_customer_019 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw339' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_020;
create regular tablespace ts_customer_020 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw340' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_021;
create regular tablespace ts_customer_021 pagesize 4K
  managed by database
  using

```

```

  (
    device '/dev/raw/raw341' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_022;
create regular tablespace ts_customer_022 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw342' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_023;
create regular tablespace ts_customer_023 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw343' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_024;
create regular tablespace ts_customer_024 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw344' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_025;
create regular tablespace ts_customer_025 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw345' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;

```

```

drop tablespace ts_customer_026;
create regular tablespace ts_customer_026 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw346' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_027;
create regular tablespace ts_customer_027 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw347' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_028;
create regular tablespace ts_customer_028 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw348' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_029;
create regular tablespace ts_customer_029 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw349' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_030;
create regular tablespace ts_customer_030 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw350' 8000000
  )
  extentsize 32

```

```

  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_031;
create regular tablespace ts_customer_031 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw351' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_032;
create regular tablespace ts_customer_032 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw352' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_033;
create regular tablespace ts_customer_033 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw353' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_034;
create regular tablespace ts_customer_034 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw354' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_035;
create regular tablespace ts_customer_035 pagesize 4K
  managed by database
  using
  (

```

```

    device '/dev/raw/raw355' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_036;
create regular tablespace ts_customer_036 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw356' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_037;
create regular tablespace ts_customer_037 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw357' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_038;
create regular tablespace ts_customer_038 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw358' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_039;
create regular tablespace ts_customer_039 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw359' 8000000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_customer_040;
create regular tablespace ts_customer_040 pagesize 4K

```

```

managed by database
using
(
    device '/dev/raw/raw360' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

drop tablespace ts_customer_041;
create regular tablespace ts_customer_041 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw361' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_042;
create regular tablespace ts_customer_042 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw362' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_043;
create regular tablespace ts_customer_043 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw363' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_044;
create regular tablespace ts_customer_044 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw364' 8000000
)
extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_045;
create regular tablespace ts_customer_045 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw365' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_046;
create regular tablespace ts_customer_046 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw366' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_047;
create regular tablespace ts_customer_047 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw367' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_048;
create regular tablespace ts_customer_048 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw368' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_049;
create regular tablespace ts_customer_049 pagesize 4K
managed by database
using
(

```

```

    device '/dev/raw/raw369' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_050;
create regular tablespace ts_customer_050 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw370' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_051;
create regular tablespace ts_customer_051 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw371' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_052;
create regular tablespace ts_customer_052 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw372' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_053;
create regular tablespace ts_customer_053 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw373' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_054;
create regular tablespace ts_customer_054 pagesize 4K

```

```

managed by database
using
(
    device '/dev/raw/raw374' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_055;
create regular tablespace ts_customer_055 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw375' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_056;
create regular tablespace ts_customer_056 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw376' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_057;
create regular tablespace ts_customer_057 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw377' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_058;
create regular tablespace ts_customer_058 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw378' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;

```

```

commit;
drop tablespace ts_customer_059;
create regular tablespace ts_customer_059 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw379' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_060;
create regular tablespace ts_customer_060 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw380' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_061;
create regular tablespace ts_customer_061 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw381' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_062;
create regular tablespace ts_customer_062 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw382' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_063;
create regular tablespace ts_customer_063 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw383' 8000000
)

```

```

)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_064;
create regular tablespace ts_customer_064 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw384' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_065;
create regular tablespace ts_customer_065 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw385' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_066;
create regular tablespace ts_customer_066 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw386' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_067;
create regular tablespace ts_customer_067 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw387' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_068;
create regular tablespace ts_customer_068 pagesize 4K
managed by database

```

```

using
(
    device '/dev/raw/raw388' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_069;
create regular tablespace ts_customer_069 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw389' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_070;
create regular tablespace ts_customer_070 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw390' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_071;
create regular tablespace ts_customer_071 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw391' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_072;
create regular tablespace ts_customer_072 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw392' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

```

```

drop tablespace ts_customer_073;
create regular tablespace ts_customer_073 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw393' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_074;
create regular tablespace ts_customer_074 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw394' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_075;
create regular tablespace ts_customer_075 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw395' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_076;
create regular tablespace ts_customer_076 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw396' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_077;
create regular tablespace ts_customer_077 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw397' 8000000
)
extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_078;
create regular tablespace ts_customer_078 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw398' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_079;
create regular tablespace ts_customer_079 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw399' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_customer_080;
create regular tablespace ts_customer_080 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw400' 8000000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

drop tablespace is_customer_001;
create regular tablespace is_customer_001 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw401' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_002;
create regular tablespace is_customer_002 pagesize 8K

```

```

managed by database
using
(
    device '/dev/raw/raw402' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_003;
create regular tablespace is_customer_003 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw403' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_004;
create regular tablespace is_customer_004 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw404' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_005;
create regular tablespace is_customer_005 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw405' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_006;
create regular tablespace is_customer_006 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw406' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;

```

```

commit;
drop tablespace is_customer_007;
create regular tablespace is_customer_007 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw407' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_008;
create regular tablespace is_customer_008 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw408' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_009;
create regular tablespace is_customer_009 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw409' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_010;
create regular tablespace is_customer_010 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw410' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_011;
create regular tablespace is_customer_011 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw411' 300000
)

```

```

extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_012;
create regular tablespace is_customer_012 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw412' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_013;
create regular tablespace is_customer_013 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw413' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_014;
create regular tablespace is_customer_014 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw414' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_015;
create regular tablespace is_customer_015 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw415' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_016;
create regular tablespace is_customer_016 pagesize 8K
managed by database
using

```



```

(
  device '/dev/raw/raw416' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_017;
create regular tablespace is_customer_017 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw417' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_018;
create regular tablespace is_customer_018 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw418' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_019;
create regular tablespace is_customer_019 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw419' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_020;
create regular tablespace is_customer_020 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw420' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_021;

```

```

create regular tablespace is_customer_021 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw421' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_022;
create regular tablespace is_customer_022 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw422' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_023;
create regular tablespace is_customer_023 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw423' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_024;
create regular tablespace is_customer_024 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw424' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_025;
create regular tablespace is_customer_025 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw425' 300000
)
extentsize 32

```

```

bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_026;
create regular tablespace is_customer_026 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw426' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_027;
create regular tablespace is_customer_027 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw427' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_028;
create regular tablespace is_customer_028 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw428' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_029;
create regular tablespace is_customer_029 pagesize 8K
managed by database
using
(
  device '/dev/raw/raw429' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_030;
create regular tablespace is_customer_030 pagesize 8K
managed by database
using
(

```

```

        device '/dev/raw/raw430' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_031;
create regular tablespace is_customer_031 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw431' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_032;
create regular tablespace is_customer_032 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw432' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_033;
create regular tablespace is_customer_033 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw433' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_034;
create regular tablespace is_customer_034 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw434' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_035;
create regular tablespace is_customer_035 pagesize 8K

```

```

    managed by database
    using
    (
        device '/dev/raw/raw435' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_036;
create regular tablespace is_customer_036 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw436' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_037;
create regular tablespace is_customer_037 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw437' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_038;
create regular tablespace is_customer_038 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw438' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_039;
create regular tablespace is_customer_039 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw439' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;

```

```

commit;
drop tablespace is_customer_040;
create regular tablespace is_customer_040 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw440' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;

drop tablespace is_customer_041;
create regular tablespace is_customer_041 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw441' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_042;
create regular tablespace is_customer_042 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw442' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_043;
create regular tablespace is_customer_043 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw443' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_044;
create regular tablespace is_customer_044 pagesize 8K
    managed by database
    using
    (

```

```

        device '/dev/raw/raw444' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_045;
create regular tablespace is_customer_045 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw445' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_046;
create regular tablespace is_customer_046 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw446' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_047;
create regular tablespace is_customer_047 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw447' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_048;
create regular tablespace is_customer_048 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw448' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_049;
create regular tablespace is_customer_049 pagesize 8K

```

```

    managed by database
    using
    (
        device '/dev/raw/raw449' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_050;
create regular tablespace is_customer_050 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw450' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_051;
create regular tablespace is_customer_051 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw451' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_052;
create regular tablespace is_customer_052 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw452' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_053;
create regular tablespace is_customer_053 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw453' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;

```

```

commit;
drop tablespace is_customer_054;
create regular tablespace is_customer_054 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw454' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_055;
create regular tablespace is_customer_055 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw455' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_056;
create regular tablespace is_customer_056 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw456' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_057;
create regular tablespace is_customer_057 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw457' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_058;
create regular tablespace is_customer_058 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw458' 300000
    )

```

```

    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_059;
create regular tablespace is_customer_059 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw459' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_060;
create regular tablespace is_customer_060 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw460' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_061;
create regular tablespace is_customer_061 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw461' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_062;
create regular tablespace is_customer_062 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw462' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_063;
create regular tablespace is_customer_063 pagesize 8K
    managed by database

```

```

    using
    (
        device '/dev/raw/raw463' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_064;
create regular tablespace is_customer_064 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw464' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_065;
create regular tablespace is_customer_065 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw465' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_066;
create regular tablespace is_customer_066 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw466' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_067;
create regular tablespace is_customer_067 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw467' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;

```

```

drop tablespace is_customer_068;
create regular tablespace is_customer_068 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw468' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_069;
create regular tablespace is_customer_069 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw469' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_070;
create regular tablespace is_customer_070 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw470' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_071;
create regular tablespace is_customer_071 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw471' 300000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_customer_072;
create regular tablespace is_customer_072 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw472' 300000
    )
    extentsize 32

```

```

bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_073;
create regular tablespace is_customer_073 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw473' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_074;
create regular tablespace is_customer_074 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw474' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_075;
create regular tablespace is_customer_075 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw475' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_076;
create regular tablespace is_customer_076 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw476' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_077;
create regular tablespace is_customer_077 pagesize 8K
managed by database
using
(

```

```

    device '/dev/raw/raw477' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_078;
create regular tablespace is_customer_078 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw478' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_079;
create regular tablespace is_customer_079 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw479' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_customer_080;
create regular tablespace is_customer_080 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw480' 300000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;

drop tablespace ts_history_001;
create regular tablespace ts_history_001 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw481' 200000
)
extentsize 32

```

```

bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_002;
create regular tablespace ts_history_002 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw482' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_003;
create regular tablespace ts_history_003 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw483' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_004;
create regular tablespace ts_history_004 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw484' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_005;
create regular tablespace ts_history_005 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw485' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_006;
create regular tablespace ts_history_006 pagesize 16K
managed by database
using
(

```

```

        device '/dev/raw/raw486' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_007;
create regular tablespace ts_history_007 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw487' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_008;
create regular tablespace ts_history_008 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw488' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_009;
create regular tablespace ts_history_009 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw489' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_010;
create regular tablespace ts_history_010 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw490' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_011;
create regular tablespace ts_history_011 pagesize 16K

```

```

    managed by database
    using
    (
        device '/dev/raw/raw491' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_012;
create regular tablespace ts_history_012 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw492' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_013;
create regular tablespace ts_history_013 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw493' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_014;
create regular tablespace ts_history_014 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw494' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_015;
create regular tablespace ts_history_015 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw495' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;

```

```

commit;
drop tablespace ts_history_016;
create regular tablespace ts_history_016 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw496' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_017;
create regular tablespace ts_history_017 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw497' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_018;
create regular tablespace ts_history_018 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw498' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_019;
create regular tablespace ts_history_019 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw499' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_020;
create regular tablespace ts_history_020 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw500' 200000
    )

```

```

    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_021;
create regular tablespace ts_history_021 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw501' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_022;
create regular tablespace ts_history_022 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw502' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_023;
create regular tablespace ts_history_023 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw503' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_024;
create regular tablespace ts_history_024 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw504' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_025;
create regular tablespace ts_history_025 pagesize 16K
    managed by database

```

```

    using
    (
        device '/dev/raw/raw505' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_026;
create regular tablespace ts_history_026 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw506' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_027;
create regular tablespace ts_history_027 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw507' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_028;
create regular tablespace ts_history_028 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw508' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_029;
create regular tablespace ts_history_029 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw509' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_history_030;
create regular tablespace ts_history_030 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw510' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_031;
create regular tablespace ts_history_031 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw511' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_032;
create regular tablespace ts_history_032 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw512' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_033;
create regular tablespace ts_history_033 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw513' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

drop tablespace ts_history_034;
create regular tablespace ts_history_034 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw514' 200000
    )
    extentsize 32

```

```

bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_035;
create regular tablespace ts_history_035 pagesize 16K
managed by database
using
(
device '/dev/raw/raw515' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_036;
create regular tablespace ts_history_036 pagesize 16K
managed by database
using
(
device '/dev/raw/raw516' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_037;
create regular tablespace ts_history_037 pagesize 16K
managed by database
using
(
device '/dev/raw/raw517' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_038;
create regular tablespace ts_history_038 pagesize 16K
managed by database
using
(
device '/dev/raw/raw518' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_039;
create regular tablespace ts_history_039 pagesize 16K
managed by database
using
(

```

```

device '/dev/raw/raw519' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_040;
create regular tablespace ts_history_040 pagesize 16K
managed by database
using
(
device '/dev/raw/raw520' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_041;
create regular tablespace ts_history_041 pagesize 16K
managed by database
using
(
device '/dev/raw/raw521' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_042;
create regular tablespace ts_history_042 pagesize 16K
managed by database
using
(
device '/dev/raw/raw522' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_043;
create regular tablespace ts_history_043 pagesize 16K
managed by database
using
(
device '/dev/raw/raw523' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;

```

```

drop tablespace ts_history_044;
create regular tablespace ts_history_044 pagesize 16K
managed by database
using
(
device '/dev/raw/raw524' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_045;
create regular tablespace ts_history_045 pagesize 16K
managed by database
using
(
device '/dev/raw/raw525' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_046;
create regular tablespace ts_history_046 pagesize 16K
managed by database
using
(
device '/dev/raw/raw526' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_047;
create regular tablespace ts_history_047 pagesize 16K
managed by database
using
(
device '/dev/raw/raw527' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_048;
create regular tablespace ts_history_048 pagesize 16K
managed by database
using
(
device '/dev/raw/raw528' 200000
)
extentsize 32

```



```

bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_049;
create regular tablespace ts_history_049 pagesize 16K
managed by database
using
(
device '/dev/raw/raw529' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_050;
create regular tablespace ts_history_050 pagesize 16K
managed by database
using
(
device '/dev/raw/raw530' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_051;
create regular tablespace ts_history_051 pagesize 16K
managed by database
using
(
device '/dev/raw/raw531' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_052;
create regular tablespace ts_history_052 pagesize 16K
managed by database
using
(
device '/dev/raw/raw532' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_053;
create regular tablespace ts_history_053 pagesize 16K
managed by database
using
(

```

```

device '/dev/raw/raw533' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_054;
create regular tablespace ts_history_054 pagesize 16K
managed by database
using
(
device '/dev/raw/raw534' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_055;
create regular tablespace ts_history_055 pagesize 16K
managed by database
using
(
device '/dev/raw/raw535' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_056;
create regular tablespace ts_history_056 pagesize 16K
managed by database
using
(
device '/dev/raw/raw536' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_057;
create regular tablespace ts_history_057 pagesize 16K
managed by database
using
(
device '/dev/raw/raw537' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_058;
create regular tablespace ts_history_058 pagesize 16K

```

```

managed by database
using
(
device '/dev/raw/raw538' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_059;
create regular tablespace ts_history_059 pagesize 16K
managed by database
using
(
device '/dev/raw/raw539' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_060;
create regular tablespace ts_history_060 pagesize 16K
managed by database
using
(
device '/dev/raw/raw540' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_061;
create regular tablespace ts_history_061 pagesize 16K
managed by database
using
(
device '/dev/raw/raw541' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_062;
create regular tablespace ts_history_062 pagesize 16K
managed by database
using
(
device '/dev/raw/raw542' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K

```

```

    prefetchsize 4096;
commit;
drop tablespace ts_history_063;
create regular tablespace ts_history_063 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw543' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_064;
create regular tablespace ts_history_064 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw544' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_065;
create regular tablespace ts_history_065 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw545' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_066;
create regular tablespace ts_history_066 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw546' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_067;
create regular tablespace ts_history_067 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw547' 200000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_068;
create regular tablespace ts_history_068 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw548' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_069;
create regular tablespace ts_history_069 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw549' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_070;
create regular tablespace ts_history_070 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw550' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_071;
create regular tablespace ts_history_071 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw551' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_072;
create regular tablespace ts_history_072 pagesize 16K
    managed by database

```

```

    using
    (
        device '/dev/raw/raw552' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_073;
create regular tablespace ts_history_073 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw553' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_074;
create regular tablespace ts_history_074 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw554' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_075;
create regular tablespace ts_history_075 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw555' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;
drop tablespace ts_history_076;
create regular tablespace ts_history_076 pagesize 16K
    managed by database
    using
    (
        device '/dev/raw/raw556' 200000
    )
    extentsize 32
    bufferpool IBMDEFAULT16K
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_history_077;
create regular tablespace ts_history_077 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw557' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_078;
create regular tablespace ts_history_078 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw558' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_079;
create regular tablespace ts_history_079 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw559' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;
drop tablespace ts_history_080;
create regular tablespace ts_history_080 pagesize 16K
managed by database
using
(
    device '/dev/raw/raw560' 200000
)
extentsize 32
bufferpool IBMDEFAULT16K
prefetchsize 4096;
commit;

drop tablespace ts_order_001;
create regular tablespace ts_order_001 pagesize 8K
managed by database
using

```

```

(
    device '/dev/raw/raw561' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_002;
create regular tablespace ts_order_002 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw562' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_003;
create regular tablespace ts_order_003 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw563' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_004;
create regular tablespace ts_order_004 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw564' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_005;
create regular tablespace ts_order_005 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw565' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_006;

```

```

create regular tablespace ts_order_006 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw566' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_007;
create regular tablespace ts_order_007 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw567' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_008;
create regular tablespace ts_order_008 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw568' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_009;
create regular tablespace ts_order_009 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw569' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_010;
create regular tablespace ts_order_010 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw570' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K

```

```

    prefetchsize 4096;
commit;
drop tablespace ts_order_011;
create regular tablespace ts_order_011 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw571' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_012;
create regular tablespace ts_order_012 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw572' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_013;
create regular tablespace ts_order_013 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw573' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_014;
create regular tablespace ts_order_014 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw574' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_015;
create regular tablespace ts_order_015 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw575' 260000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_016;
create regular tablespace ts_order_016 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw576' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_017;
create regular tablespace ts_order_017 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw577' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_018;
create regular tablespace ts_order_018 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw578' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_019;
create regular tablespace ts_order_019 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw579' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_020;
create regular tablespace ts_order_020 pagesize 8K
    managed by database

```

```

    using
    (
        device '/dev/raw/raw580' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_021;
create regular tablespace ts_order_021 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw581' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_022;
create regular tablespace ts_order_022 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw582' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_023;
create regular tablespace ts_order_023 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw583' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_order_024;
create regular tablespace ts_order_024 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw584' 260000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;

```

```

commit;
drop tablespace ts_order_025;
create regular tablespace ts_order_025 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw585' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_026;
create regular tablespace ts_order_026 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw586' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_027;
create regular tablespace ts_order_027 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw587' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_028;
create regular tablespace ts_order_028 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw588' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_029;
create regular tablespace ts_order_029 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw589' 260000
  )

```

```

  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_030;
create regular tablespace ts_order_030 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw590' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_031;
create regular tablespace ts_order_031 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw591' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_032;
create regular tablespace ts_order_032 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw592' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_033;
create regular tablespace ts_order_033 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw593' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_034;
create regular tablespace ts_order_034 pagesize 8K
  managed by database
  using

```

```

  (
    device '/dev/raw/raw594' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_035;
create regular tablespace ts_order_035 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw595' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_036;
create regular tablespace ts_order_036 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw596' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_037;
create regular tablespace ts_order_037 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw597' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_038;
create regular tablespace ts_order_038 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw598' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_039;

```

```

create regular tablespace ts_order_039 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw599' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_040;
create regular tablespace ts_order_040 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw600' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;

drop tablespace ts_order_041;
create regular tablespace ts_order_041 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw601' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_042;
create regular tablespace ts_order_042 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw602' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_043;
create regular tablespace ts_order_043 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw603' 260000
  )

```

```

  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_044;
create regular tablespace ts_order_044 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw604' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_045;
create regular tablespace ts_order_045 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw605' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_046;
create regular tablespace ts_order_046 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw606' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_047;
create regular tablespace ts_order_047 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw607' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_048;
create regular tablespace ts_order_048 pagesize 8K
  managed by database
  using

```

```

  (
    device '/dev/raw/raw608' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_049;
create regular tablespace ts_order_049 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw609' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_050;
create regular tablespace ts_order_050 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw610' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_051;
create regular tablespace ts_order_051 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw611' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_052;
create regular tablespace ts_order_052 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw612' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_053;

```

```

create regular tablespace ts_order_053 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw613' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_054;
create regular tablespace ts_order_054 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw614' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_055;
create regular tablespace ts_order_055 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw615' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_056;
create regular tablespace ts_order_056 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw616' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_057;
create regular tablespace ts_order_057 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw617' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K

```

```

  prefetchsize 4096;
commit;
drop tablespace ts_order_058;
create regular tablespace ts_order_058 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw618' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_059;
create regular tablespace ts_order_059 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw619' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_060;
create regular tablespace ts_order_060 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw620' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_061;
create regular tablespace ts_order_061 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw621' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_062;
create regular tablespace ts_order_062 pagesize 8K
  managed by database
  using
  (

```

```

    device '/dev/raw/raw622' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_063;
create regular tablespace ts_order_063 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw623' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_064;
create regular tablespace ts_order_064 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw624' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_065;
create regular tablespace ts_order_065 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw625' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_066;
create regular tablespace ts_order_066 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw626' 260000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_order_067;
create regular tablespace ts_order_067 pagesize 8K

```

```

managed by database
using
(
    device '/dev/raw/raw627' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_068;
create regular tablespace ts_order_068 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw628' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_069;
create regular tablespace ts_order_069 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw629' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_070;
create regular tablespace ts_order_070 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw630' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_071;
create regular tablespace ts_order_071 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw631' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;

```

```

commit;
drop tablespace ts_order_072;
create regular tablespace ts_order_072 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw632' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_073;
create regular tablespace ts_order_073 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw633' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_074;
create regular tablespace ts_order_074 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw634' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_075;
create regular tablespace ts_order_075 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw635' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_076;
create regular tablespace ts_order_076 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw636' 260000
)

```

```

extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_077;
create regular tablespace ts_order_077 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw637' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_078;
create regular tablespace ts_order_078 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw638' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_079;
create regular tablespace ts_order_079 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw639' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_order_080;
create regular tablespace ts_order_080 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw640' 260000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;

```



```

drop tablespace is_order_001;
create regular tablespace is_order_001 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw641' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_002;
create regular tablespace is_order_002 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw642' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_003;
create regular tablespace is_order_003 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw643' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_004;
create regular tablespace is_order_004 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw644' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_005;
create regular tablespace is_order_005 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw645' 220000
)
extentsize 32

```

```

bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_006;
create regular tablespace is_order_006 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw646' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_007;
create regular tablespace is_order_007 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw647' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_008;
create regular tablespace is_order_008 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw648' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_009;
create regular tablespace is_order_009 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw649' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_010;
create regular tablespace is_order_010 pagesize 8K
managed by database
using
(

```

```

    device '/dev/raw/raw650' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_011;
create regular tablespace is_order_011 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw651' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_012;
create regular tablespace is_order_012 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw652' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_013;
create regular tablespace is_order_013 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw653' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_014;
create regular tablespace is_order_014 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw654' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_015;
create regular tablespace is_order_015 pagesize 8K

```

```

managed by database
using
(
    device '/dev/raw/raw655' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_016;
create regular tablespace is_order_016 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw656' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_017;
create regular tablespace is_order_017 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw657' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_018;
create regular tablespace is_order_018 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw658' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_019;
create regular tablespace is_order_019 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw659' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;

```

```

commit;
drop tablespace is_order_020;
create regular tablespace is_order_020 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw660' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_021;
create regular tablespace is_order_021 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw661' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_022;
create regular tablespace is_order_022 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw662' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_023;
create regular tablespace is_order_023 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw663' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_024;
create regular tablespace is_order_024 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw664' 220000
)

```

```

)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_025;
create regular tablespace is_order_025 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw665' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_026;
create regular tablespace is_order_026 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw666' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_027;
create regular tablespace is_order_027 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw667' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_028;
create regular tablespace is_order_028 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw668' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_029;
create regular tablespace is_order_029 pagesize 8K
managed by database

```

```

using
(
    device '/dev/raw/raw669' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_030;
create regular tablespace is_order_030 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw670' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_031;
create regular tablespace is_order_031 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw671' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_032;
create regular tablespace is_order_032 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw672' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_033;
create regular tablespace is_order_033 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw673' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;

```

```

drop tablespace is_order_034;
create regular tablespace is_order_034 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw674' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_035;
create regular tablespace is_order_035 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw675' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_036;
create regular tablespace is_order_036 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw676' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_037;
create regular tablespace is_order_037 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw677' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_038;
create regular tablespace is_order_038 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw678' 220000
)
extentsize 32

```

```

bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_039;
create regular tablespace is_order_039 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw679' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_040;
create regular tablespace is_order_040 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw680' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;

drop tablespace is_order_041;
create regular tablespace is_order_041 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw681' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_042;
create regular tablespace is_order_042 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw682' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_043;
create regular tablespace is_order_043 pagesize 8K
managed by database

```

```

using
(
    device '/dev/raw/raw683' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_044;
create regular tablespace is_order_044 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw684' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_045;
create regular tablespace is_order_045 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw685' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_046;
create regular tablespace is_order_046 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw686' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_047;
create regular tablespace is_order_047 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw687' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;

```

```

drop tablespace is_order_048;
create regular tablespace is_order_048 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw688' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_049;
create regular tablespace is_order_049 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw689' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_050;
create regular tablespace is_order_050 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw690' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_051;
create regular tablespace is_order_051 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw691' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_052;
create regular tablespace is_order_052 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw692' 220000
)
extentsize 32

```

```

bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_053;
create regular tablespace is_order_053 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw693' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_054;
create regular tablespace is_order_054 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw694' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_055;
create regular tablespace is_order_055 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw695' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_056;
create regular tablespace is_order_056 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw696' 220000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace is_order_057;
create regular tablespace is_order_057 pagesize 8K
managed by database
using
(

```

```

        device '/dev/raw/raw697' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_058;
create regular tablespace is_order_058 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw698' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_059;
create regular tablespace is_order_059 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw699' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_060;
create regular tablespace is_order_060 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw700' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_061;
create regular tablespace is_order_061 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw701' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_062;

```

```

create regular tablespace is_order_062 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw702' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_063;
create regular tablespace is_order_063 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw703' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_064;
create regular tablespace is_order_064 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw704' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_065;
create regular tablespace is_order_065 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw705' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_066;
create regular tablespace is_order_066 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw706' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K

```

```

    prefetchsize 4096;
commit;
drop tablespace is_order_067;
create regular tablespace is_order_067 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw707' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_068;
create regular tablespace is_order_068 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw708' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_069;
create regular tablespace is_order_069 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw709' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_070;
create regular tablespace is_order_070 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw710' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_071;
create regular tablespace is_order_071 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw711' 220000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_072;
create regular tablespace is_order_072 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw712' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_073;
create regular tablespace is_order_073 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw713' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_074;
create regular tablespace is_order_074 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw714' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_075;
create regular tablespace is_order_075 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw715' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_076;
create regular tablespace is_order_076 pagesize 8K
    managed by database

```

```

    using
    (
        device '/dev/raw/raw716' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_077;
create regular tablespace is_order_077 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw717' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_078;
create regular tablespace is_order_078 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw718' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_079;
create regular tablespace is_order_079 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw719' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace is_order_080;
create regular tablespace is_order_080 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw720' 220000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_orderline_001;
create regular tablespace ts_orderline_001 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw721' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_002;
create regular tablespace ts_orderline_002 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw722' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_003;
create regular tablespace ts_orderline_003 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw723' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_004;
create regular tablespace ts_orderline_004 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw724' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_005;
create regular tablespace ts_orderline_005 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw725' 6600000
    )

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_006;
create regular tablespace ts_orderline_006 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw726' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_007;
create regular tablespace ts_orderline_007 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw727' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_008;
create regular tablespace ts_orderline_008 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw728' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_009;
create regular tablespace ts_orderline_009 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw729' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_010;
create regular tablespace ts_orderline_010 pagesize 8K
managed by database

```

```

using
(
    device '/dev/raw/raw730' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_011;
create regular tablespace ts_orderline_011 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw731' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_012;
create regular tablespace ts_orderline_012 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw732' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_013;
create regular tablespace ts_orderline_013 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw733' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_014;
create regular tablespace ts_orderline_014 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw734' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_orderline_015;
create regular tablespace ts_orderline_015 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw735' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_016;
create regular tablespace ts_orderline_016 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw736' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_017;
create regular tablespace ts_orderline_017 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw737' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_018;
create regular tablespace ts_orderline_018 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw738' 6600000
)
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_019;
create regular tablespace ts_orderline_019 pagesize 8K
managed by database
using
(
    device '/dev/raw/raw739' 6600000
)
    extentsize 32

```

```

bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_020;
create regular tablespace ts_orderline_020 pagesize 8K
managed by database
using
(
device '/dev/raw/raw740' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_021;
create regular tablespace ts_orderline_021 pagesize 8K
managed by database
using
(
device '/dev/raw/raw741' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_022;
create regular tablespace ts_orderline_022 pagesize 8K
managed by database
using
(
device '/dev/raw/raw742' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_023;
create regular tablespace ts_orderline_023 pagesize 8K
managed by database
using
(
device '/dev/raw/raw743' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_024;
create regular tablespace ts_orderline_024 pagesize 8K
managed by database
using

```

```

(
device '/dev/raw/raw744' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_025;
create regular tablespace ts_orderline_025 pagesize 8K
managed by database
using
(
device '/dev/raw/raw745' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_026;
create regular tablespace ts_orderline_026 pagesize 8K
managed by database
using
(
device '/dev/raw/raw746' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_027;
create regular tablespace ts_orderline_027 pagesize 8K
managed by database
using
(
device '/dev/raw/raw747' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_028;
create regular tablespace ts_orderline_028 pagesize 8K
managed by database
using
(
device '/dev/raw/raw748' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_029;

```

```

create regular tablespace ts_orderline_029 pagesize 8K
managed by database
using
(
device '/dev/raw/raw749' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_030;
create regular tablespace ts_orderline_030 pagesize 8K
managed by database
using
(
device '/dev/raw/raw750' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_031;
create regular tablespace ts_orderline_031 pagesize 8K
managed by database
using
(
device '/dev/raw/raw751' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_032;
create regular tablespace ts_orderline_032 pagesize 8K
managed by database
using
(
device '/dev/raw/raw752' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K
prefetchsize 4096;
commit;
drop tablespace ts_orderline_033;
create regular tablespace ts_orderline_033 pagesize 8K
managed by database
using
(
device '/dev/raw/raw753' 6600000
)
extentsize 32
bufferpool IBMDEFAULT8K

```



```

    prefetchsize 4096;
commit;
drop tablespace ts_orderline_034;
create regular tablespace ts_orderline_034 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw754' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_035;
create regular tablespace ts_orderline_035 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw755' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_036;
create regular tablespace ts_orderline_036 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw756' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_037;
create regular tablespace ts_orderline_037 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw757' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_038;
create regular tablespace ts_orderline_038 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw758' 6600000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_039;
create regular tablespace ts_orderline_039 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw759' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_040;
create regular tablespace ts_orderline_040 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw760' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_041;
create regular tablespace ts_orderline_041 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw761' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_042;
create regular tablespace ts_orderline_042 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw762' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_043;

```

```

create regular tablespace ts_orderline_043 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw763' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_044;
create regular tablespace ts_orderline_044 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw764' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_045;
create regular tablespace ts_orderline_045 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw765' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_046;
create regular tablespace ts_orderline_046 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw766' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_047;
create regular tablespace ts_orderline_047 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw767' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K

```

```

        prefetchsize 4096;
commit;
drop tablespace ts_orderline_048;
create regular tablespace ts_orderline_048 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw768' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_049;
create regular tablespace ts_orderline_049 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw769' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_050;
create regular tablespace ts_orderline_050 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw770' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_051;
create regular tablespace ts_orderline_051 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw771' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_052;
create regular tablespace ts_orderline_052 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw772' 6600000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_053;
create regular tablespace ts_orderline_053 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw773' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_054;
create regular tablespace ts_orderline_054 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw774' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_055;
create regular tablespace ts_orderline_055 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw775' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_056;
create regular tablespace ts_orderline_056 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw776' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_057;
create regular tablespace ts_orderline_057 pagesize 8K
    managed by database

```

```

    using
    (
        device '/dev/raw/raw777' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_058;
create regular tablespace ts_orderline_058 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw778' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_059;
create regular tablespace ts_orderline_059 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw779' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_060;
create regular tablespace ts_orderline_060 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw780' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;
drop tablespace ts_orderline_061;
create regular tablespace ts_orderline_061 pagesize 8K
    managed by database
    using
    (
        device '/dev/raw/raw781' 6600000
    )
    extentsize 32
    bufferpool IBMDEFAULT8K
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_orderline_062;
create regular tablespace ts_orderline_062 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw782' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_063;
create regular tablespace ts_orderline_063 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw783' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_064;
create regular tablespace ts_orderline_064 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw784' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_065;
create regular tablespace ts_orderline_065 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw785' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_066;
create regular tablespace ts_orderline_066 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw786' 6600000
  )

```

```

  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_067;
create regular tablespace ts_orderline_067 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw787' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_068;
create regular tablespace ts_orderline_068 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw788' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_069;
create regular tablespace ts_orderline_069 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw789' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_070;
create regular tablespace ts_orderline_070 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw790' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_071;
create regular tablespace ts_orderline_071 pagesize 8K
  managed by database
  using

```

```

  (
    device '/dev/raw/raw791' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_072;
create regular tablespace ts_orderline_072 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw792' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_073;
create regular tablespace ts_orderline_073 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw793' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_074;
create regular tablespace ts_orderline_074 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw794' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_075;
create regular tablespace ts_orderline_075 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw795' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_076;

```

```

create regular tablespace ts_orderline_076 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw796' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_077;
create regular tablespace ts_orderline_077 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw797' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_078;
create regular tablespace ts_orderline_078 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw798' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_079;
create regular tablespace ts_orderline_079 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw799' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K
  prefetchsize 4096;
commit;
drop tablespace ts_orderline_080;
create regular tablespace ts_orderline_080 pagesize 8K
  managed by database
  using
  (
    device '/dev/raw/raw800' 6600000
  )
  extentsize 32
  bufferpool IBMDEFAULT8K

```

```

  prefetchsize 4096;
commit;
drop tablespace ts_newordA_001;
create regular tablespace ts_newordA_001 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw801' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_002;
create regular tablespace ts_newordA_002 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw802' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_003;
create regular tablespace ts_newordA_003 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw803' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_004;
create regular tablespace ts_newordA_004 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw804' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_005;

```

```

create regular tablespace ts_newordA_005 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw805' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_006;
create regular tablespace ts_newordA_006 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw806' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_007;
create regular tablespace ts_newordA_007 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw807' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_008;
create regular tablespace ts_newordA_008 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw808' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_009;
create regular tablespace ts_newordA_009 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw809' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP

```

```

    prefetchsize 4096;
commit;
drop tablespace ts_newordA_010;
create regular tablespace ts_newordA_010 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw810' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_011;
create regular tablespace ts_newordA_011 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw811' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_012;
create regular tablespace ts_newordA_012 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw812' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_013;
create regular tablespace ts_newordA_013 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw813' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_014;
create regular tablespace ts_newordA_014 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw814' 130000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordA_015;
create regular tablespace ts_newordA_015 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw815' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_016;
create regular tablespace ts_newordA_016 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw816' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_017;
create regular tablespace ts_newordA_017 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw817' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_018;
create regular tablespace ts_newordA_018 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw818' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_019;
create regular tablespace ts_newordA_019 pagesize 4K
  managed by database

```

```

  using
  (
    device '/dev/raw/raw819' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_020;
create regular tablespace ts_newordA_020 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw820' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_021;
create regular tablespace ts_newordA_021 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw821' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_022;
create regular tablespace ts_newordA_022 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw822' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_023;
create regular tablespace ts_newordA_023 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw823' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;

```

```

commit;
drop tablespace ts_newordA_024;
create regular tablespace ts_newordA_024 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw824' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_025;
create regular tablespace ts_newordA_025 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw825' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_026;
create regular tablespace ts_newordA_026 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw826' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_027;
create regular tablespace ts_newordA_027 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw827' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_028;
create regular tablespace ts_newordA_028 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw828' 130000
  )

```

```

  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_029;
create regular tablespace ts_newordA_029 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw829' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_030;
create regular tablespace ts_newordA_030 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw830' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_031;
create regular tablespace ts_newordA_031 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw831' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_032;
create regular tablespace ts_newordA_032 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw832' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_033;
create regular tablespace ts_newordA_033 pagesize 4K
  managed by database
  using

```

```

  (
    device '/dev/raw/raw833' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_034;
create regular tablespace ts_newordA_034 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw834' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_035;
create regular tablespace ts_newordA_035 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw835' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_036;
create regular tablespace ts_newordA_036 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw836' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_037;
create regular tablespace ts_newordA_037 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw837' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_038;

```

```

create regular tablespace ts_newordA_038 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw838' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_039;
create regular tablespace ts_newordA_039 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw839' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_040;
create regular tablespace ts_newordA_040 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw840' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

drop tablespace ts_newordA_041;
create regular tablespace ts_newordA_041 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw841' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_042;
create regular tablespace ts_newordA_042 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw842' 130000
)

```

```

extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_043;
create regular tablespace ts_newordA_043 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw843' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_044;
create regular tablespace ts_newordA_044 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw844' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_045;
create regular tablespace ts_newordA_045 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw845' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_046;
create regular tablespace ts_newordA_046 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw846' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_047;
create regular tablespace ts_newordA_047 pagesize 4K
managed by database
using

```

```

(
    device '/dev/raw/raw847' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_048;
create regular tablespace ts_newordA_048 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw848' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_049;
create regular tablespace ts_newordA_049 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw849' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_050;
create regular tablespace ts_newordA_050 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw850' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_051;
create regular tablespace ts_newordA_051 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw851' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_052;

```

```

create regular tablespace ts_newordA_052 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw852' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_053;
create regular tablespace ts_newordA_053 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw853' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_054;
create regular tablespace ts_newordA_054 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw854' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_055;
create regular tablespace ts_newordA_055 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw855' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_056;
create regular tablespace ts_newordA_056 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw856' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP

```

```

  prefetchsize 4096;
commit;
drop tablespace ts_newordA_057;
create regular tablespace ts_newordA_057 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw857' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_058;
create regular tablespace ts_newordA_058 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw858' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_059;
create regular tablespace ts_newordA_059 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw859' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_060;
create regular tablespace ts_newordA_060 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw860' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_061;
create regular tablespace ts_newordA_061 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw861' 130000

```

```

  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_062;
create regular tablespace ts_newordA_062 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw862' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_063;
create regular tablespace ts_newordA_063 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw863' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_064;
create regular tablespace ts_newordA_064 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw864' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_065;
create regular tablespace ts_newordA_065 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw865' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordA_066;
create regular tablespace ts_newordA_066 pagesize 4K

```



```

managed by database
using
(
    device '/dev/raw/raw866' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_067;
create regular tablespace ts_newordA_067 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw867' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_068;
create regular tablespace ts_newordA_068 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw868' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_069;
create regular tablespace ts_newordA_069 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw869' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_070;
create regular tablespace ts_newordA_070 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw870' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;

```

```

commit;
drop tablespace ts_newordA_071;
create regular tablespace ts_newordA_071 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw871' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_072;
create regular tablespace ts_newordA_072 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw872' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_073;
create regular tablespace ts_newordA_073 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw873' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_074;
create regular tablespace ts_newordA_074 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw874' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_075;
create regular tablespace ts_newordA_075 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw875' 130000
)

```

```

extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_076;
create regular tablespace ts_newordA_076 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw876' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_077;
create regular tablespace ts_newordA_077 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw877' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_078;
create regular tablespace ts_newordA_078 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw878' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_079;
create regular tablespace ts_newordA_079 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw879' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordA_080;
create regular tablespace ts_newordA_080 pagesize 4K
managed by database
using

```

```

(
  device '/dev/raw/raw880' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

drop tablespace ts_newordB_001;
create regular tablespace ts_newordB_001 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw881' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_002;
create regular tablespace ts_newordB_002 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw882' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_003;
create regular tablespace ts_newordB_003 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw883' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_004;
create regular tablespace ts_newordB_004 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw884' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;

```

```

drop tablespace ts_newordB_005;
create regular tablespace ts_newordB_005 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw885' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_006;
create regular tablespace ts_newordB_006 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw886' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_007;
create regular tablespace ts_newordB_007 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw887' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_008;
create regular tablespace ts_newordB_008 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw888' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_009;
create regular tablespace ts_newordB_009 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw889' 130000
)
extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_010;
create regular tablespace ts_newordB_010 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw890' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_011;
create regular tablespace ts_newordB_011 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw891' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_012;
create regular tablespace ts_newordB_012 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw892' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_013;
create regular tablespace ts_newordB_013 pagesize 4K
managed by database
using
(
  device '/dev/raw/raw893' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_014;
create regular tablespace ts_newordB_014 pagesize 4K
managed by database
using
(

```

```

        device '/dev/raw/raw894' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_015;
create regular tablespace ts_newordB_015 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw895' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_016;
create regular tablespace ts_newordB_016 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw896' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_017;
create regular tablespace ts_newordB_017 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw897' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_018;
create regular tablespace ts_newordB_018 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw898' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_019;
create regular tablespace ts_newordB_019 pagesize 4K

```

```

    managed by database
    using
    (
        device '/dev/raw/raw899' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_020;
create regular tablespace ts_newordB_020 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw900' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_021;
create regular tablespace ts_newordB_021 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw901' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_022;
create regular tablespace ts_newordB_022 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw902' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_023;
create regular tablespace ts_newordB_023 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw903' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP

```

```

    prefetchsize 4096;
commit;
drop tablespace ts_newordB_024;
create regular tablespace ts_newordB_024 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw904' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_025;
create regular tablespace ts_newordB_025 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw905' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_026;
create regular tablespace ts_newordB_026 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw906' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_027;
create regular tablespace ts_newordB_027 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw907' 130000
    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_028;
create regular tablespace ts_newordB_028 pagesize 4K
    managed by database
    using
    (
        device '/dev/raw/raw908' 130000

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_029;
create regular tablespace ts_newordB_029 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw909' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_030;
create regular tablespace ts_newordB_030 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw910' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_031;
create regular tablespace ts_newordB_031 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw911' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_032;
create regular tablespace ts_newordB_032 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw912' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_033;
create regular tablespace ts_newordB_033 pagesize 4K
managed by database

```

```

using
(
    device '/dev/raw/raw913' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_034;
create regular tablespace ts_newordB_034 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw914' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_035;
create regular tablespace ts_newordB_035 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw915' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_036;
create regular tablespace ts_newordB_036 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw916' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_037;
create regular tablespace ts_newordB_037 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw917' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_newordB_038;
create regular tablespace ts_newordB_038 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw918' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_039;
create regular tablespace ts_newordB_039 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw919' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_040;
create regular tablespace ts_newordB_040 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw920' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

drop tablespace ts_newordB_041;
create regular tablespace ts_newordB_041 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw921' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_042;
create regular tablespace ts_newordB_042 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw922' 130000
)

```

```

    )
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_043;
create regular tablespace ts_newordB_043 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw923' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_044;
create regular tablespace ts_newordB_044 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw924' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_045;
create regular tablespace ts_newordB_045 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw925' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_046;
create regular tablespace ts_newordB_046 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw926' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_047;
create regular tablespace ts_newordB_047 pagesize 4K
managed by database

```

```

using
(
    device '/dev/raw/raw927' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_048;
create regular tablespace ts_newordB_048 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw928' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_049;
create regular tablespace ts_newordB_049 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw929' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_050;
create regular tablespace ts_newordB_050 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw930' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_051;
create regular tablespace ts_newordB_051 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw931' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;

```

```

drop tablespace ts_newordB_052;
create regular tablespace ts_newordB_052 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw932' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_053;
create regular tablespace ts_newordB_053 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw933' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_054;
create regular tablespace ts_newordB_054 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw934' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_055;
create regular tablespace ts_newordB_055 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw935' 130000
)
    extentsize 32
    bufferpool IBMDEFAULTBP
    prefetchsize 4096;
commit;
drop tablespace ts_newordB_056;
create regular tablespace ts_newordB_056 pagesize 4K
managed by database
using
(
    device '/dev/raw/raw936' 130000
)
    extentsize 32

```

```

bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_057;
create regular tablespace ts_newordB_057 pagesize 4K
managed by database
using
(
device '/dev/raw/raw937' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_058;
create regular tablespace ts_newordB_058 pagesize 4K
managed by database
using
(
device '/dev/raw/raw938' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_059;
create regular tablespace ts_newordB_059 pagesize 4K
managed by database
using
(
device '/dev/raw/raw939' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_060;
create regular tablespace ts_newordB_060 pagesize 4K
managed by database
using
(
device '/dev/raw/raw940' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_061;
create regular tablespace ts_newordB_061 pagesize 4K
managed by database
using
(

```

```

device '/dev/raw/raw941' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_062;
create regular tablespace ts_newordB_062 pagesize 4K
managed by database
using
(
device '/dev/raw/raw942' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_063;
create regular tablespace ts_newordB_063 pagesize 4K
managed by database
using
(
device '/dev/raw/raw943' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_064;
create regular tablespace ts_newordB_064 pagesize 4K
managed by database
using
(
device '/dev/raw/raw944' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_065;
create regular tablespace ts_newordB_065 pagesize 4K
managed by database
using
(
device '/dev/raw/raw945' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_066;

```

```

create regular tablespace ts_newordB_066 pagesize 4K
managed by database
using
(
device '/dev/raw/raw946' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_067;
create regular tablespace ts_newordB_067 pagesize 4K
managed by database
using
(
device '/dev/raw/raw947' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_068;
create regular tablespace ts_newordB_068 pagesize 4K
managed by database
using
(
device '/dev/raw/raw948' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_069;
create regular tablespace ts_newordB_069 pagesize 4K
managed by database
using
(
device '/dev/raw/raw949' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP
prefetchsize 4096;
commit;
drop tablespace ts_newordB_070;
create regular tablespace ts_newordB_070 pagesize 4K
managed by database
using
(
device '/dev/raw/raw950' 130000
)
extentsize 32
bufferpool IBMDEFAULTBP

```

```

prefetchsize 4096;
commit;
drop tablespace ts_newordB_071;
create regular tablespace ts_newordB_071 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw951' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_072;
create regular tablespace ts_newordB_072 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw952' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_073;
create regular tablespace ts_newordB_073 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw953' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_074;
create regular tablespace ts_newordB_074 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw954' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_075;
create regular tablespace ts_newordB_075 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw955' 130000

```

```

)
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_076;
create regular tablespace ts_newordB_076 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw956' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_077;
create regular tablespace ts_newordB_077 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw957' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_078;
create regular tablespace ts_newordB_078 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw958' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_079;
create regular tablespace ts_newordB_079 pagesize 4K
  managed by database
  using
  (
    device '/dev/raw/raw959' 130000
  )
  extentsize 32
  bufferpool IBMDEFAULTBP
  prefetchsize 4096;
commit;
drop tablespace ts_newordB_080;
create regular tablespace ts_newordB_080 pagesize 4K
  managed by database

```

```

  using
  (
    device '/dev/raw/raw960' 130000
  )
  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 6400;
alter bufferpool WDS2 size 6400;
alter bufferpool WDS3 size 6400;
alter bufferpool WDS4 size 6400;
alter bufferpool WDS5 size 6400;
alter bufferpool WDS6 size 6400;
alter bufferpool WDS7 size 6400;
alter bufferpool WDS8 size 6400;
alter bufferpool CST1 size 9000;
alter bufferpool CST2 size 9000;
alter bufferpool CST3 size 9000;
alter bufferpool CST4 size 9000;
alter bufferpool CST5 size 9000;
alter bufferpool CST6 size 9000;
alter bufferpool CST7 size 9000;
alter bufferpool CST8 size 9000;
alter bufferpool CSTI1 size 300000;
alter bufferpool CSTI2 size 300000;
alter bufferpool CSTI3 size 300000;
alter bufferpool CSTI4 size 300000;
alter bufferpool CSTI5 size 300000;
alter bufferpool CSTI6 size 300000;
alter bufferpool CSTI7 size 300000;
alter bufferpool CSTI8 size 300000;
alter bufferpool NEW1 size 318400;
alter bufferpool NEW2 size 318400;
alter bufferpool NEW3 size 318400;
alter bufferpool NEW4 size 318400;
alter bufferpool NEW5 size 318400;
alter bufferpool NEW6 size 318400;
alter bufferpool NEW7 size 318400;
alter bufferpool NEW8 size 318400;
alter bufferpool HST1 size 16800;
alter bufferpool HST2 size 16800;
alter bufferpool HST3 size 16800;

```

```

alter bufferpool HST4 size 16800;
alter bufferpool HST5 size 16800;
alter bufferpool HST6 size 16800;
alter bufferpool HST7 size 16800;
alter bufferpool HST8 size 16800;
alter bufferpool ORD1 size 180000;
alter bufferpool ORD2 size 180000;
alter bufferpool ORD3 size 180000;
alter bufferpool ORD4 size 180000;
alter bufferpool ORD5 size 180000;
alter bufferpool ORD6 size 180000;
alter bufferpool ORD7 size 180000;
alter bufferpool ORD8 size 180000;
alter bufferpool IORD1 size 510000;
alter bufferpool IORD2 size 510000;
alter bufferpool IORD3 size 510000;
alter bufferpool IORD4 size 510000;
alter bufferpool IORD5 size 510000;
alter bufferpool IORD6 size 510000;
alter bufferpool IORD7 size 510000;
alter bufferpool IORD8 size 510000;
alter bufferpool OLN1 size 310000;
alter bufferpool OLN2 size 310000;
alter bufferpool OLN3 size 310000;
alter bufferpool OLN4 size 310000;
alter bufferpool OLN5 size 310000;
alter bufferpool OLN6 size 310000;
alter bufferpool OLN7 size 310000;
alter bufferpool OLN8 size 310000;
alter bufferpool STK1 size 12562528;
alter bufferpool STK2 size 12562528;
alter bufferpool STK3 size 12562528;
alter bufferpool STK4 size 12562528;
alter bufferpool STK5 size 12562528;
alter bufferpool STK6 size 12562528;
alter bufferpool STK7 size 12562528;
alter bufferpool STK8 size 12562528;
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
LD_FLAGS_EXEC=
LD_FLAGS_SHLIB=shared
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB)
LD_FLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2 -m64
LD_FLAGS_OUT=-o

# Library Configuration
AR=ar
AR_FLAGS=-rv
AR_FLAGS_LIB=
AR_FLAGS_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 - 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

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 = tpcctdbg$(OBJEXT)
UTIL_OBJ_GEN = tpcctmisc$(OBJEXT)
UTIL_OBJ_DB2 = tpcctctx$(OBJEXT)

#
#####
#####
# User Targets
#
#####
#####
all: $(UTIL_OBJ_DBG) $(UTIL_OBJ_GEN) connect $(UTIL_OBJ_DB2)
disconnect

dbgen: $(UTIL_OBJ_GEN)

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

#
#####
#####
# Helper Targets
#
#####
#####
connect:
- db2 connect to $(TPCC_DBNAME)

disconnect:
- db2 connect reset
- db2 terminate

rebind: connect
db2 bind tpcctctx.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/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 - 2006
** 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 = $(TPCC_ROOT)/Src.Common/tpccclwh$(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 "tpccrmd.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%\n";
char fmtDist[] = "%d%04.4f%.2f%s%s%s%s%s%s%04.4f%.2f%\n";
char fmtItem[] = "%s%.2f%s%04.4f%.2f%\n";
char fmtStock[] = "%d%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%\n";
char fmtCust[] = "%d%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%\n";
char fmtHist[] = "%d%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%\n";
char fmtOrdr[] = "%d%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%\n";
char fmtOLine[] = "%s%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%\n";
char fmtNewOrd[] = "%d%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%04.4f%.2f%\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 necessary
    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 necessary. */
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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    if (!quiet_mode) {
        fprintf(stdout, "\nWAREHOUSE table generated in %8.2f
seconds.\n\n", elapsed);
        fflush(stdout);
    }
} else {
    fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after %8.2f
seconds.\n\n", ware_num, elapsed);
    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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    if (!quiet_mode) {
        fprintf(stdout, "\nDISTRICT table generated in %8.2f seconds.\n\n", elapsed);
        fflush(stdout);
    }
} else {
}

} else {

```

```

        fprintf(stderr, "\nDISTRICT table FAILED at (W %d D %d) after %8.2f
seconds.\n\n", ware_num, dist_num, elapsed);
        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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    if (!quiet_mode) {
        fprintf(stdout,"nCUSTOMER table generated in %8.2f seconds.\n\n",elapsed);
        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, elapsed);
        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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    if (!quiet_mode) {
        fprintf(stdout,"nHISTORY table generated in %8.2f seconds.\n\n",elapsed);
        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, elapsed);
        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; }

    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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    if (!quiet_mode) {
        fprintf(stdout, "\nNEW_ORDER table generated in %8.2f
seconds.\n\n", elapsed);
        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, elapsed);
    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", 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();
    elapsed = timestamp2 - timestamp1;
    if (rc1 == 0 && rc2 == 0) {
        if (!quiet_mode) {
            fprintf(stdout, "\nORDERS & ORDER_LINE tables generated in %8.2f
seconds.\n\n", elapsed);
            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, elapsed);
        fflush(stderr);
    }
}

// This routine will initialize 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 "ival.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);
    random(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);
}

/*
*****
* NUrnd_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.
*/
/* *****
*/

#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_L_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 96000
#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,
            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 - Miscellaneous 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 authorization 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



August 7, 2008

IBM Corporation
Ray Venditti
System x 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 9.5 Lic/1 year			
Maintenance	2400	278.52	668,448
SW Maintenance Renewal - 2 year	4800	13.27	63,696
		Sub-total reference price for DB2 ESE:	732,144
		TOTAL REFERENCE PRICE:	732,144


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,

A handwritten signature in black ink that reads "Richard Hughes".

Richard Hughes
IBM Sales & Distribution, Software Sales
Americas Sales Executive DB2 and Informix
212-493-2065
rhughes@us.ibm.com


CDW CDW.G CDW Canada 866.756.4239
Shopping Cart 0 Items Support Log On



Shop CDW

My Account

Search for... All Products [Browse All Categories](#)

Print This Page 

Products



NO IMAGE AVAILABLE

[View Larger](#)

[Send To An Associate](#) >
[Find Similar Products](#) >

D-Link

D-Link Unmanaged (24) Port 10/100/1000Mbps Layer 2 Switch

Product ID
 CDW Part: 674865
 Mfg. Part: DGS-1024D
 UNSPSC: 43222612



Availability:
 In stock and ready for shipment
 . Ships same day if ordered before 4 p.m. CT



Product Pricing



Price: **\$209.99**

Qty:



Recommended Accessories

 Belkin 8 Outlet, 6' Cord Surge **\$14.95** 

 Tripp Lite 10' Blue Cat5e or Cat5 RJ45 Molded 350mhz UTP Patch Cable **\$4.99** 

 Tripp Lite 10' Blue Cat6 Gigabit Snagless Patch **\$7.99** 

 Belkin 14' 650MHZ Certified CAT 6 Patch Cable Blue **\$16.99** 

 Black Box 10' Cat5e GigaBase 350 Patch Cable, Purple **\$5.00** 

Overview [Specs](#) [Accessories](#)

Main Features

- 24-port 10/100/1000BASE-T (RJ-45) Gigabit Ethernet unmanaged rackmountable switch

Specifications are provided by the manufacturer. Refer to the manufacturer for an explanation of the print speed and other ratings.

High-speed, rack mount or desktop switch

Get the blazing speed of Gigabit Ethernet with the D-Link DGS-1024D, a 24-port 10/100/1000Mbps Switch that delivers power, performance, and reliability in one cost-effective, space-saving design. Increase the speed of your network server and backbone connections, or make Gigabit to the desktop a reality. Power users in the office, workgroup, or creative production environment can now move large, bandwidth-intensive files faster. Transfer graphics, CGI, CAD, or multimedia files across the network instantly. The versatile and compact 1RU height design of the DGS-1024D enables the device to be rackmounted in a standard 19 inch rack, while conserving valuable rack space.

It features a non-blocking switching architecture that filters and forwards packets at full wire-speed for maximum throughput. An 8K MAC address table provides scalability for even the largest networks. Address learning and aging, 802.3x Flow Control for full-duplex mode, and back pressure flow control for half-duplex mode alleviates traffic congestion and ensures reliable data transmission. Designed using industry standards, it is compatible with virtually all 10, 100, and 1000Mbps Ethernet devices and other vendor equipment. It protects your existing network

CDW CDW.G CDW Canada 800.756.4239 [Shopping Cart](#) [0 Items](#) [Support](#) [Log On](#)

Shop CDW [Print This Page](#)

Products **My Account** Search for... [All Products](#) [Find It](#) [Browse All Categories](#)

Services: [Planning](#) [Implementation](#) [Support](#) [Hosted Services](#) [Managed Services](#)

Links KVM ports **Product Pricing**
Price: \$41.99
 Qty: [Add to cart](#)

[View Larger](#)
[Send To An Associate](#) [Find Similar Products](#)

Product ID
 CDW Part: 430446
 Mfg. Part: KVM2KIT
 UNSPSC: 43211604

Availability:
 In stock and ready for shipment
 . Ships same day if ordered
 before 4 p.m. CT

Overview **Specs**

Main Features

- KVM switch
- 2 ports
- 1 local user

Specifications are provided by the manufacturer. Refer to the manufacturer for an explanation of the print speed and other ratings.

Easily Control Two PCs from a Single Keyboard, Mouse, and Monitor

The Linksys ProConnect Integrated KVM 2-Port Switch lets you take instant command over two PCs - all from a single keyboard, mouse, and monitor! For your convenience, it even comes with built-in cables to meet all your connection needs. Switch between PCs with a quick, two-key action, or use the automatic scan feature to easily monitor both computers.

The KVM Switch eliminates those pesky, cable-swapping duties while saving you desktop space and hardware costs. Because it doesn't use software, the KVM Switch is compatible with all major desktop and notebook computers. It's the perfect choice for the home, office, or anywhere else you work with two PCs at the same time.

[About Us](#) [Careers](#) [Newsroom](#) [Terms and Conditions](#) [Contact Us](#)

The Right Technology. Right Away. 

Copyright © 2008 CDW Corporation

[Cart](#) | [Register](#) | [Log In](#)



- [Home](#)
- [Solutions](#)
- [Services & Products](#)
- [Partners](#)
- [Developers](#)
- [Training](#)
- [Support](#)
- [Store](#)

- [Order History](#)
- [View Saved Carts](#)
- [Renew Subscriptions](#)

Shopping Cart

	Item	Quantity	Price	Line Total
<p>New Subscription Contract July 30, 2008 - July 30, 2011</p>				
	Red Hat Enterprise Linux Advanced Platform for IBM POWER, Premium (unlimited sockets) - for 3 Years July 30, 2008 - July 30, 2011	<input type="text" value="1"/>	Remove	\$6,747.00 \$6,747.00
Promotion Code: <input type="text"/>			Update Cart	Subtotal: \$6,747.00

Optional Install Discs and Documentation

[add to cart](#) Red Hat Enterprise Linux 5 (for POWER) \$25 Media Kit (DVD only)

[Continue shopping](#)

[Continue to Checkout](#)

[ABOUT SSL CERTIFICATES](#)



U: P: [Login](#)

SPEAK WITH A STORAGE EXPERT NOW
866.463.3372

Search

Advanced Search

Shopping Cart

1 x SB5600-16A... \$4,390.00

sub-total: \$4,390.00

Empty View Cart Checkout

- Products**
- HDS (Hitachi) Storage
 - EMC Storage
 - NetApp Storage
 - Nexsan Storage
 - Silon Storage
 - Fabric Switches
 - Brocade
 - QLogic
 - SANbox 1400
 - SANbox 5200
 - SANbox 5202
 - SANbox 5600**
 - SANbox 5600Q
 - SANbox 5602
 - SANbox 5602V
 - SANbox 5602V
 - SANbox 5600
 - SAN Directors
 - HBA's & Adapters
 - 8Gb Fibre Channel
 - 4Gb Fibre Channel
 - 2Gb Fibre Channel
 - 1Gb iSCSI
 - 20Gb InfiniBand
 - 10Gb InfiniBand
 - 10Gb FCoE
 - 1Gb Ethernet
 - Multiprotocol Routers
 - InfiniBand Switches
 - Cables
 - Tape / VTL Libraries
 - Transceivers
 - Appliances
 - Software
 - Open Box Inventory
 - End of Life

Manufacturers

Please Select

BRCCADE [CLICK FOR DETAILS](#)

Silkworm 5000 Fabric Switch
 16 or 32 Ports at 4Gb/sec.

Home | Contact Us | About Us | Privacy Policy | Conditions of Use | Conditions of Sale

Products - Fabric Switches - QLogic - SANbox 5600 - SB5600-16A-E



SANbox 5600 16-Port 4Gb Fabric Switch [SFPs included]

MSRP: \$9,394.00
 Save: -\$4,944.00

SANDirect \$4,390.00

Qty: [Add to Cart](#) VeriSign® Secured



Model: SB5600-16A-E

Overview: QLogic SANbox 5600 fibre switch with (16) 4Gb ports enabled and (1) power supply. SFPs Included.

SB5600-16A-E

- View Related Products
- View all QLOGIC SANbox 5600 4Gb Enterprise Fabric Switches
- Bookmark this Page

CHAT WITH A LIVE STORAGE ARCHITECT

EMA® Email Archiver
 Sarbanes-Oxley Compliant
 AES Encrypted - Digital Signatures
 Uses Existing iSCSI, NAS & FC Storage

Now the benefits of stackable IP switches are available for high performance 4Gb SANs

The SANbox 5600 stackable switch delivers the seamless scalability and performance of a chassis switch, in a simple-to-manage, pay-as-you-grow solution. With sixteen 4Gb ports plus a four-pack of high-speed 10Gb ISL ports, and entry as low as 8 ports with 4-port software-keyed increments and included graphical user interface (GUI) wizards, each SANbox 5600 stackable switch provides maximum flexibility for configuring, managing and scaling SANs.

Start small, then expand as your needs grow.

Entry as low as eight ports, then scale in four-port software-keyed increments. Each chassis includes sixteen 4Gb ports, plus a four-pack of high-speed 10Gb ISL ports. Built-in graphical user interface (GUI) with installation/configuration wizards. Available in two power supply configurations: single integrated (SB5600) and dual hot-swappable (SB5602).

Key Features

- 8, 12 or 16 auto detecting 4Gb/2Gb/1Gb device ports
- 4 10Gb ports for high speed stacking links
- 4-port 4Gb/2Gb/1Gb and/or 10Gb field upgrade licenses available
- Stacking of up to 6 units for 96 available user ports
- Non-Disruptive Code Load and Activation (NDCLA)
- Single and dual hot-swap power supply configurations available.
- Configuration, Zoning and Extended Distance wizards to simplify switch installation and fabric scaling
- Interoperable with all FC-SW-2 compliant Fibre Channel switches
- Full-fabric, public-loop or switch-to-switch connectivity on 2Gb/1Gb ports
- Full-fabric or switch-to-switch connectivity on 10Gb ports
- Auto-sensing, self-configuring ports
- Fabric tracking tool for fabric-wide snapshots and detection of configuration changes
- Non-blocking full-bandwidth architecture
- RSCN suppression
- "No-Wait" routing - guaranteed maximum performance independent of data traffic
- Industry's lowest latency for maximum performance
- SFP (small form-factor pluggable) connectivity - 16 front ports in a 1U full-width rack form-factor
- Seamless operation with higher-level third-party management applications
- In-band, out-of-band, Telnet and SNMP management access
- ASIC-embedded memory - faster, more scalable and more reliable than shared memory architecture

Breakthrough Ease of Use

The first Fibre Channel stackable switches to be configured and zoned with simple wizards. Stack management is simple throughout your SAN lifecycle as you install, configure, monitor, diagnose and upgrade your QLogic HBAs and switches.

Incredibly Low Cost

Starting at eight ports, SANbox 5600 stackable switches offer an entry point affordable to the smallest business. Managing your stack is so simple, you won't have to hire a SAN expert.

Modular Scalability

Start with 8 ports and then grow your SAN in 4-port increments up to 96 ports in a single stack. A key benefit is the ability to add, change or delete switches without disrupting your storage network.

Document Library

- QLogic SANbox 5600 Series Data Sheet
- Switch Management User's Guide - 5.0
- Switch Installation Guide - 5.0
- SANbox Quick Start Guide

System Configuration

Max. Port Count: 20
 Min. Port Count: 8
 Port Upgrades: 4
 Link Speed Support: 1Gb, 2Gb, 4Gb
 Redundant Power Supplies: No
 Form Factor: 1U

HBAs, Transceivers and Fibre Directors

QLOGIC

SANblade QLE2460-CK 4Gb PCIe
 \$879.00

EMULEX

Lightpulse LP9002S-E 2Gb SBus [BMC Firmware]
 \$1,189.00



Home > My Shopping Cart

MY SHOPPING CART

My Wish Lists | Print Cart | Email Cart

Update Qty's Remove Selected Move Selected To...

Qty.	Product Description	Savings	Total Price
<input type="checkbox"/>	powercom KIN-1500AP 1500VA 900 Watts UPS - Retail Item #: N82E16842106115 Return Policy: Standard Return Policy	-\$15.00 Instant	\$149.99 \$134.99
<input type="checkbox"/> Protect Your Investment (expand for options)			

Subtotal: \$134.99

Calculate Shipping

Zip Code: UPS Guaranteed 3 Day Service

Shipping: \$0.00

Redeem Gift Certificates

Claim Code:
Security Code:

Gift Certificates: \$0.00

Apply Promo Code

Promo Code:

Promo Code: \$0.00

Grand Total:* \$134.99

* Above total does not include shipping or taxes. Please input zip code to calculate your grand total.

Having problems with your cart? [Click here for help](#) or [try emptying your cart to start over.](#)

* [Click here to view important shipping information.](#)

[Policy & Agreement](#) | [Privacy Policy](#) | © 2000-2008 Newegg Inc. All rights reserved.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

August 8, 2008

IBM Corporation
Chris King
3079 Cornwallis Road
Durham, NC 27709

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
127-00012	Visual Studio Standard 2005 <i>Full License No Discount Applied</i>	\$250	1	\$250
N/A	Microsoft Problem Resolution Services <i>Professional Support (1 Incident)</i>	\$245	1	\$245

Windows Server 2008 and Windows Server 2003 are currently orderable through Microsoft's normal distribution channels. A list of Microsoft's resellers can be found at <http://www.microsoft.com/products/info/render.aspx?view=22&type=mpn&content=22/licensing>

SQL Server 2008 will be orderable and available by August 30, 2008.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$245 per call.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCchki0808080000006834.
Please include this Reference ID in any correspondence regarding this price quote.

