

---

**IBM Power 780 Server**  
**Model 9179-MHB**

*Using*

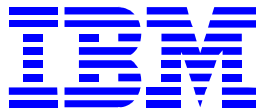
**AIX Version 6.1**

*and*

**DB2 Enterprise 9.5**

---

**TPC Benchmark<sup>TM</sup> C**  
**Full Disclosure Report**



Second Edition

April 19, 2010

### *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 Power

AIX

IBM

DB2, DB2 Enterprise 9.5

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 2008 server and COM+ are registered trademarks of Microsoft Corporation

**First Edition: April 13, 2010**

**Second Edition: April 19, 2010**

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.

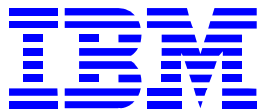
Request for additional copies of this document should be sent to the following address:

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, 2010. 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.

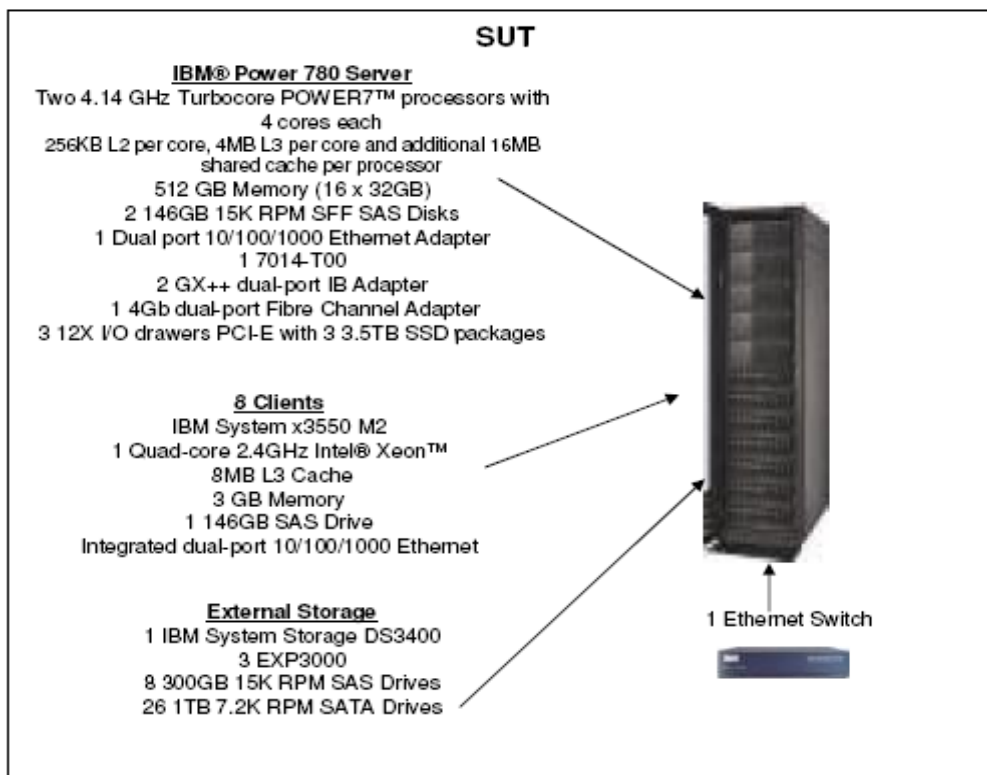


**IBM Power 780  
Model 9179-MHB**

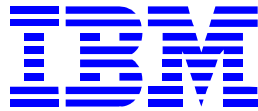
TPC-C Rev. 5.10.1

Report Date: April 13, 2010

<b>Total System Cost</b>	<b>TPC-C Throughput</b>	<b>Price/Performance</b>	<b>Availability Date</b>	
\$825,004 USD	<b>1,200,011</b>	\$0.69 USD	<b>October 13, 2010</b>	
<b>Database Server Processors/Cores/Threads</b>	<b>Database Manager</b>	<b>Operating System</b>	<b>Other Software</b>	<b>No. Users</b>
2/8/32 POWER7 4.14GHz	DB2 9.5	AIX V6.1	Microsoft Visual C++ Microsoft COM+	962,370



System Components	Server		Each of the 8 Clients	
	Quantity	Description	Quantity	Description
<b>Processors /Cores/Threads</b>	2/8/32	4.14GHz POWER7	1/4/8	2.4GHz Intel Xeon
<b>Memory</b>	16	32GB	3	1 GB
<b>Disk Controllers</b>	1	4 Gb FC Adapter	1	SAS Controller
<b>Disk Drives</b>	3 8 26 2	3.5TB SSD Package 300GB 15K RPM SAS 1TB 7.2K RPM SATA 146GB 15K RPM SFF SAS	1	146GB 10K RPM SAS
<b>Total Storage</b>		37,609GB		136.61GB
<b>Terminals</b>	1	System Console	1	System Console



## IBM Power 780 Model 9179-MHB

TPC-C Rev. 5.10.1

Report Date: April 13, 2010

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
<b>Server Hardware</b>						
Server 1:9179-MHB Base MTM	9179-MHB	1	10,195	1	10,195	8,315
IBM MMB DRWR, IBM BEZEL +CHASSIS/IBM LABELS+	5597	1	12,000	1	12,000	
AC POWER SUPPLY, 200-240V, 1725 WATT	5632	1	1,502	2	3,004	
FSP1 FLEXIBLE SERVICE PROCESSOR CARD	5664	1	4,000	1	4,000	
QUAD (2X RJ45 1GB/ 2X SFP+ 10GB) HEA	1803	1	699	1	699	
GX++ DUAL-PORT IB ADPTR	1808	1	1,499	2	2,998	
1.5 Meter 12X DDR Cable	1862	1	524	8	4,192	
SAS Cable (X) Adapter to SAS Enclosure	3661	1	197	1	197	
PWR CBL, DRWR TO IBM PDU, 9' 200-240V/10A	6671	1	19	8	152	
IBM/OEM Rack MTG RAIL kit, ADJUSTABLE DEPTH	7164	1	222	2	444	
OPERATOR PANEL + SHIP GROUP, P7 MR	1853	1	1,000	1	1,000	
3.8 / 4.1GHZ, 0/16w Core POWER7, 16 DDR3 Memory Slots	4982	1	57,429	1	57,429	6,072
1W PROCESSOR ACTIVATION FOR FC 4982	5469	1	8,375	8	67,000	23,040
0/128GB(4X32GB) SDRAM DDR3 DIMMS, 1066MHZ	5602	1	15,440	4	61,760	
1GB DDR3 MEMORY ACTIVATION	8212	1	245	512	125,440	
146GB 15K RPM SFF SAS DISK	1886	1	1,045	2	2,090	
System AC Power Supply, 1725 w	5632	1	1,652	2	3,304	
DISK/MEDIA BACK PLANE, 6X SFF DISK BAYS, 1X SA	5652	1	4,000	1	4,000	
SATA Slimline DVD-RAM Drive	5762	1	392	1	392	
12X I/O DRAWER PCIE, SFF Disk	5902	1	14,277	3	42,831	23,400
4 Gigabit Fibre Channel dual-port PCI-X Adapter (Jack)	5759	1	3,308	1	3,308	
IBM 2-port 10/100/1000 Base-TX PCI Express Adapter	5767	1	692	1	692	
Power Control Cable (SPCN) - 3 Meter	6006	1	52	4	208	
Dual Port 12X Channel Attach - Short Hun	6446	1	755	1	755	
I/O Drawer Mounting Enclosure	7314	1	687	1	687	
Front Door OEM (Black)	6246	1	970	1	970	
Side Panel (Black)	6098	1	150	2	300	
PDU to 14', 200-240V/24A, UTG0247, PT#12	6654	1	240	2	480	
Power Dist Unit-Side Mount, Universal UTG0247	7188	1	1,000	2	2,000	
HMC 1:7042-C07 Desktop Hardw.Mgmt.Console	7042-C07	1	1,830	1	1,830	1,344
IBM T117 FLAT PANEL MONITOR	3645	1	875	1	875	
Power Cord (6-foot), To Wall Plug Type #4	6470	1	18	2	36	
Ethernet Cable, 6M, HMC to System Unit	7801	1	15	1	15	
Keyboard - English, #103P	5951	1	107	1	107	
USB Mouse	8945	1	39	1	39	
3.5 TB SSD Package	FC4367	1	99,114	3	294,342	50,400
IBM 36U Enterprise Rack	7014-T00	1	2,920	1	2,920	768
DPI Single-phase 30A/208V C19 Enterprise PDU (US)	6062	1	669	2	1,338	
				<b>Subtotal</b>	<b>714,029</b>	<b>113,339</b>
<b>Storage</b>						
DS3400	1726-42E	1	9,292	1	9,292	1,300
EXP3000	1727-01X-2678	1	3,199	3	9,597	2,290
1 TB 7.2K rpm SATA 1000GB	1727-01X-5561	1	1,199	26	31,174	
300GB HS 15K SAS HDD	5532	1	699	8	5,592	
SAS Cables 0.6 Meters	3688	1	119	4	476	
Fiber Cable 1m	5601	1	79	2	158	
				<b>Subtotal</b>	<b>56,289</b>	<b>3,580</b>
<b>Server Software</b>						
AIX V6 (media only)	5692-A6P	1	50	1	50	
AIX 6 for POWER V6.1	5765-G62	1	2,600	8	20,800	
AIX per processor SWMA Large Power 7 (3Y)	5773-SM3-1260	1	1,755	8		14,040
AIX per processor SWMA Large Power 7 24x7 Upgrade (3Y)	5773-SM3-1261	1	718	8		5,744
HMC Software SUB (3Y)	5773-0570	1	461	1		461
HMC Software Support (3Y)	5773-0569	1	675	1		675
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	1,140	1	1,140	
C for AIX user annual SW maint renewal	E1A1FLL	1	228	2		456
DB2 9.5 Enterprise Proc Lic/1 year Maintenance		1	405	960	388,800	
DB2 9.5 Enterprise Edition Proc Maint Renew		1	81	1,920		155,520
				<b>Subtotal</b>	<b>410,790</b>	<b>176,896</b>



## IBM Power 780 Model 9179-MHB

TPC-C Rev. 5.10.1

Report Date: April 13, 2010

Client Hardware and Software						
IBM System x3550 M2 (Quad-core Xeon 2.4GHz)	7948AC1	1	3,316	8	26,528	
1 GB memory	3963	1	85	24	2,040	
146GB 10K RPM SAS SFF	5537	1	269	8	2,152	
Optical 3-Button Mouse - USB	8913	1	19	1	19	
Preferred Pro Full Size PS/2 Keyboard	40K9584	1	29	1	29	
ServicePac for 3-Year 24x7x4 Support	6756298	1	450	8		3,600
IBM T115 15" TFT Monitor	494215U	1	209	1	209	
			<b>Subtotal</b>		<b>30,977</b>	<b>3,600</b>
Third Party Hardware/Software						
Microsoft Visual Studio 2008 Professional	127-00166	2	799	1	799	
Windows Web Server 2008 R2	LWA-00984	2	469	8	3,752	
Microsoft Problem Resolution Services		2	259	1	259	245
3Com Baseline Switch 2824 24-port unmanaged Gigabit	1931828	3	297	1	297	
			<b>Subtotal</b>		<b>5,107</b>	<b>245</b>
<b>Total</b>					<b>1,217,192</b>	<b>297,660</b>
<b>Total IBM Discounts*</b>						<b>-689,848</b>
<b>Three-Year Cost of Ownership</b>						<b>825,004</b>
<b>tpmC</b>						<b>1,200,011</b>
<b>\$/tpmC</b>						<b>0.69</b>

### Notes:

For pricing details and contact information please see appendix D.

Pricing Sources: 1)IBM 2)Microsoft 3)CDW

\*Discounts are based on US list prices for similar quantities & configurations including pre-payment for maintenance. The discount applies to the totality of all items with price sources of "1".

Audited by: Francois Raab, Info Sizing ([www.infosizing.com](http://www.infosizing.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 the stated prices are not available according to these terms, please inform the TPC at [pricing@tpc.org](mailto:pricing@tpc.org). Thank you

## Numerical Quantities Summary for the IBM Power 780 Server Model 9179-MHB

MQTH, computed Maximum Qualified Throughput: 1,200,011 tpmC

<u>Response Times (in seconds)</u>	<u>90<sup>th</sup> %</u>	<u>Average</u>	<u>Maximum</u>
New Order	0.694	0.403	12.522
Payment	0.689	0.397	12.229
Order-Status	0.684	0.395	9.244
Delivery (interactive)	0.459	0.234	8.912
Delivery (deferred)	0.48	0.26	10.42
Stock-Level	0.719	0.421	9.801
Menu	0.464	0.236	12.411

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	44.963%
Payment	43.007%
Order-Status	4.012%
Delivery	4.008%
Stock-Level	4.010%

<u>Keying/Think Times (in seconds)</u>	<u>Min.</u>	<u>Average</u>	<u>Max.</u>
New Order	18.01/0.01	18.01/12.02	18.03/120.21
Payment	3.01/0.01	3.01/12.02	3.03/120.21
Order-Status	2.01/0.01	2.01/10.01	2.03/100.10
Delivery	2.01/0.01	2.01/5.02	2.03/50.20
Stock-Level	2.01/0.01	2.01/5.02	2.03/50.21

### Test Duration

Ramp-up Time	1 hour 11 minutes 39 secs
Measurement interval	2 hours 0 minutes
Transactions during measurement interval (all types)	320,264,245
Ramp-down time	30 minutes

### Checkpoints

Number of checkpoints	N/A
Checkpoint interval	N/A

# Table of Content

Preface .....	10
0 General Items.....	11
0.1. Application Code Disclosure .....	11
0.2. Benchmark Sponsor.....	11
0.3. Parameter Settings .....	11
0.4. Configuration Diagrams .....	11
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 .....	15
2.8. Home and Remote Payment Transactions .....	15
2.9. Non-Primary Key Transactions .....	15
2.10. Skipped Delivery Transactions.....	15
2.11. Mix of Transaction Types.....	16
2.12. Queuing Mechanism of Delivery.....	16
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 .....	20
4.4. Partitions/Replications Mapping .....	20
4.5. 60-Day Space Calculations.....	25
5 Clause 5: Performance Metrics and Response Time Related Items.....	26
5.1. Response Times.....	26
5.2. Keying and Think Times .....	26
5.3. Response Time Frequency Distribution .....	27
5.4. Performance Curve for Response Time versus Throughput.....	29
5.5. Think Time Frequency Distribution .....	30
5.6. Throughput versus Elapsed Time .....	30
5.7. Steady State Determination .....	31
5.8. Work Performed During Steady State .....	31
5.9. Measurement Interval .....	32
6 Clause 6: SUT, Driver, and Communication Definition Related Items.....	33
6.1. RTE Availability .....	33
6.2. Functionality and Performance of Emulated Components .....	33
6.3. Network Bandwidth.....	33
6.4. Operator Intervention .....	33
7 Clause 7: Pricing Related Items.....	34
7.1. Hardware and Programs Used .....	34
7.2. Three Year Cost of System Configuration .....	34
7.3. Availability Dates.....	34
7.4. Statement of tpmC and Price/Performance.....	34

7.5.	Country-specific pricing .....	34
7.6.	Orderability Date .....	34
8	Clause 9: Audit Related Items .....	36
Appendix - A:	Client Server Code .....	39
A.1	Client/Terminal Handler Code .....	39
A.2	Client Transaction Code .....	81
Appendix - B:	Tunable Parameters .....	135
B.1	Database Parameters .....	135
B.2	Transaction Monitor Parameters.....	137
B.3	AIX Parameters .....	139
Appendix - C:	Database Setup Code .....	141
C.1	Database Creation Scripts.....	141
C.2	Data Generation Code .....	460
Appendix - D:	Pricing Information .....	476



## Abstract

---

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.10.1 dated February, 2009, for measurements on the IBM Power 780 Server Model 9179-MHB. The software used on the IBM Power 780 Server Model 9179-MHB includes AIX Version 6.1 operating system, DB2 9.5 database manager. Microsoft COM+ is used as transaction manager.

### IBM Power 780 Server Model 9179-MHB

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM Power 780 Model 9179-MHB	DB2 9.5	AIX Version 6.1

Total System Cost	TPC-C Throughput	Price/Performance
<ul style="list-style-type: none"><li>• Hardware</li><li>• Software</li><li>• 3 Years Maintenance</li></ul>	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC
\$825,004 USD	1,200,011	\$0.69 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 5.10.1 in February 2009.

This is the full disclosure report for benchmark testing of the IBM Power 780 Server Model 9179-MHB 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 IBM 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.*

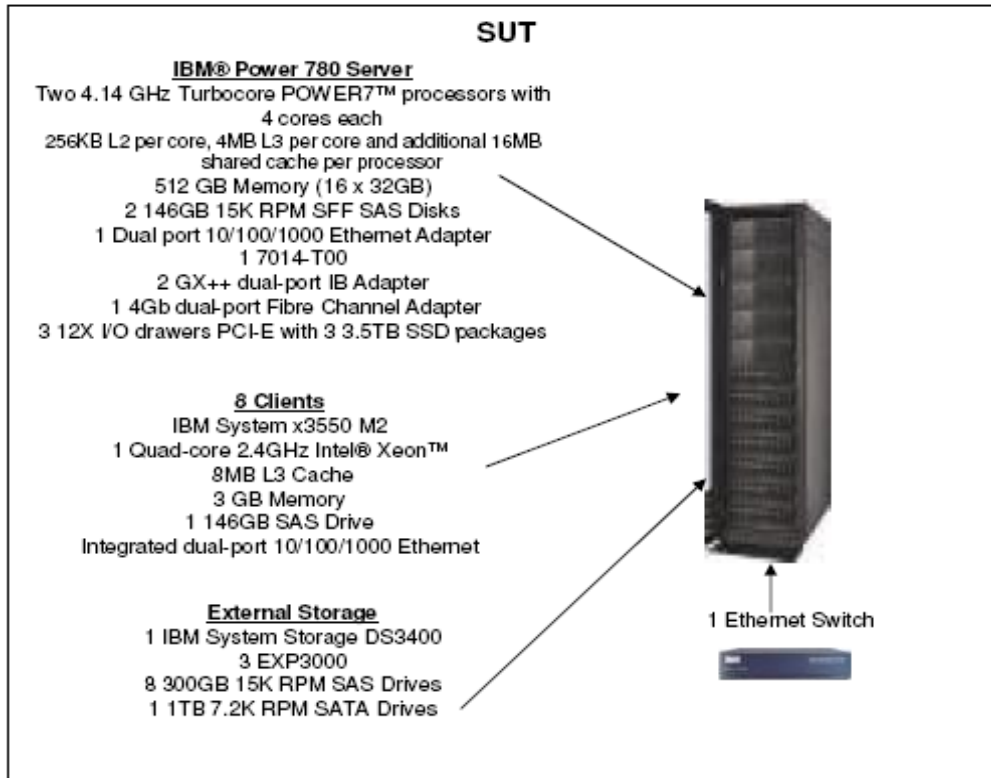
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

### 0.4. Configuration Diagrams

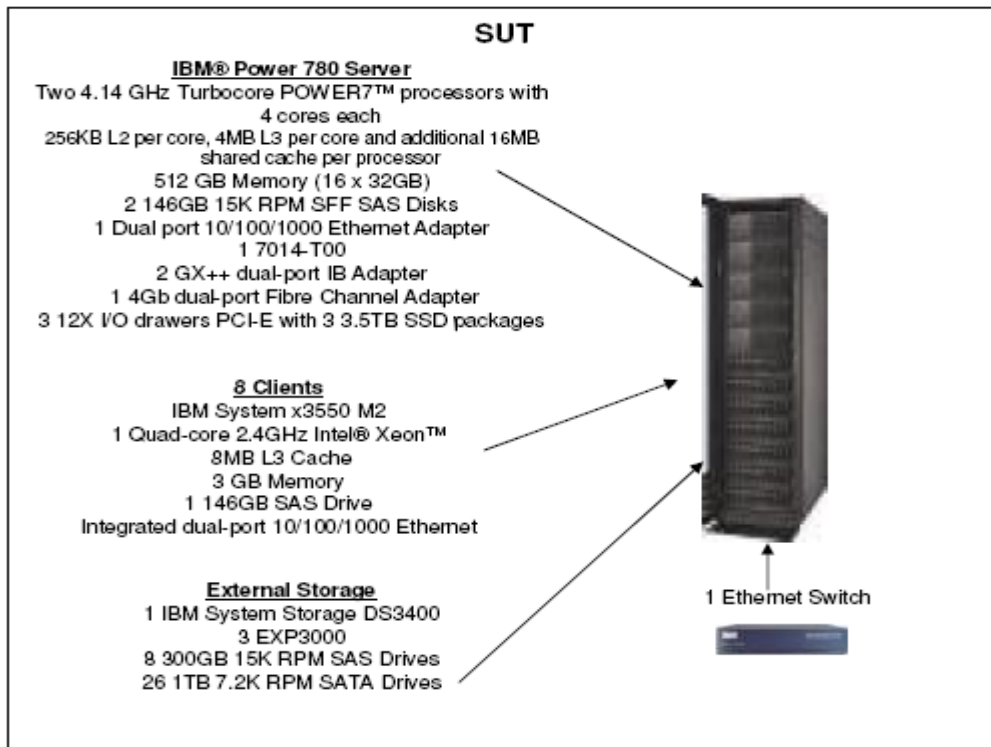
*Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:*

- *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 Power 780 Server Model 9179-MHB Benchmark Configuration



## IBM Power 780 Server Model 9179-MHB Priced Configuration



---

# 1 Clause 1: Logical Data Base Design Related Items

## 1.1. Table Definitions

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

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

## 1.2. Database Organization

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

Physical space was allocated to DB2 9.5 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 9.5 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.*

All tables but ITEM were horizontally partitioned into multiple tables.

Each table partition for WAREHOUSE, DISTRICT, STOCK, CUSTOMER, HISTORY, ORDERS, ORDERLINE and NEWORDER contains data associated with a range of 867 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 `srandom()`, `getpid()` and `gettimeofday()` functions are used to produce unique random seeds for each driver. The drivers use these seeds to seed the `srand()`, `srandom()` and `srand48()` functions. Random numbers are produced using wrappers around the standard system random number generators.

The negative exponential distribution uses the following function to generate the distribution. This function has the property of producing a negative exponential curve with a specified average and a maximum value 4 times the average.

```
const double RANDOM_4_Z = 0.89837799236185
const double RANDOM_4_K = 0.97249842407114
double neg_exp_4(double average {
    return - average * (1/RANDOM_4_Z * log (1 - RANDOM_4_K * drand48()));
})
```

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

### 2.2. Input/Output Screens

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

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

### 2.3. Priced Terminal Features

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

The emulated workstations, IBM x3550 systems, are commercially available and support all of the requirements in Clause 2.2.2.4.h

### 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.

<b>New Order</b>	<b>IBM Power 780 Model 9179-MHB</b>
Percentage of Home order lines	99.00%
Percentage of Remote order lines	1.00%
Percentage of Rolled Back Transactions	0.99%
Average Number of Items per order	10.00
<b>Payment</b>	
Percentage of Home transactions	85.00%
Percentage of Remote transactions	15.00%
<b>Non-Primary Key Access</b>	
Percentage of Payment using C_LAST	60.00%
Percentage of Order-Status using C_LAST	59.98%
<b>Delivery</b>	
Delivery transactions skipped	0
<b>Transaction Mix</b>	
New-Order	44.96%
Payment	43.01%
Order-Status	4.01%
Delivery	4.01%
Stock-Level	4.01%

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

## 2.12. Queuing Mechanism of Delivery

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

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



---

## 3 Clause 3: Transaction and System Properties

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

All ACID tests were conducted according to specification.

### 3.1. Atomicity Requirements

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

#### 3.1.1. Atomicity of Completed Transaction

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

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

1. The balance, BALANCE\_1, was retrieved from the CUSTOMER table for a random Customer, District and Warehouse combination.
2. The Payment transaction was executed and committed for the Customer, District, and Warehouse combination used in step 1.
3. The balance, BALANCE\_2, was retrieved again for the Customer, District, and Warehouse combination used in step 1 and step 2. It was verified that BALANCE\_1 was greater than BALANCE\_2 by the amount of the Payment transaction.

#### 3.1.2. Atomicity of Aborted Transactions

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

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

1. The Payment application code was implemented with a Perl script that allowed the transaction to be rolled back rather than committed.
2. The balance, BALANCE\_3, was retrieved from the Customer table for the same Customer, District, and Warehouse combination used in the completed Payment transaction Atomicity test.
3. The Payment transaction was executed for the Customer, District and Warehouse used in step 2. Rather than commit the transaction, the transaction was rolled back.
4. The balance, BALANCE\_4 was retrieved again for the Customer, District, and Warehouse combination used in step 2. It was verified that BALANCE\_4 was equal to BALANCE\_3, demonstrating that there were no remaining effects of the rolled back Payment transaction.

### 3.2. Consistency Requirements

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

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

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

1. The sum of balances (d\_ytd) for all Districts within a specific Warehouse is equal to the balance (w\_ytd) of that Warehouse.
2. For each District within a Warehouse, the next available Order ID (d\_next\_o\_id) minus one is equal to the most recent Order ID [max(o\_id)] for the Order table associated with the 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*

#### 3.4.1. Permanent Unrecoverable Failure of any Single Durable Medium

*Permanent irrecoverable failure of any single durable medium containing TPC-C data base tables or recovery log data.*

##### *Failure of Log Disk and Log Cache:*

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

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 test was started and continued to run for several minutes at a throughput around 12.5% of the reported tpmC.
3. One of the disks containing the transaction log was removed. Since the disk was RAID-5, the SUT continued to process the transactions successfully.
4. The test continued for at least another 5 minutes.
5. Since write cache mirroring was enabled for the log device, one of the RAID controllers, which holds one copy of the mirrored cache, was removed. There was a brief pause in I/O while the failover to the remaining log controller occurred. The controller detected a mirror-out-of-sync condition and deactivated write-back cache.
6. The run continued to completion without write-back cache.
7. The disk from step 3 was replaced and the controller was inserted after the completion of the run

8. Step 1 was performed returning the value for SUM\_2. It was verified that SUM\_2 was greater than or equal to SUM\_1 plus the completed New\_Order transactions recorded by the RTE..

***Failure of Durable Medium Containing TPC-C Database Tables:***

The following steps were successfully performed to demonstrate Durability against the failure of a disk unit with database tables:

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 about 12.5% of the full load . The test continued to run at about 12.5% of the reported tpmC for 6 minutes.
4. To induce the disk failure, the system was powered off.
5. One of the data disks was replaced. The system was powered back on and logical volumes were restored to functional state. The full database was restored from the backup copy in step 1.
6. The database was restarted and the transactions in the log were applied to the database.
7. Step 2 was performed returning SUM\_2. It was verified that SUM\_2 was greater than or equal to SUM\_1 plus the completed New\_Order transactions recorded by the RTE.
8. Consistency condition 3 was verified.

***Instantaneous Interruption and Memory Failure:***

The following steps were conducted on a fully scaled database:

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 continued to run for several minutes at a throughput level well above 90% of the reported tpmC.
3. The system was powered off, which removed power from all system components, including memory.
4. The system was powered back on and the database completed the recovery process.
5. Step 1 was performed returning SUM\_2. It was verified that SUM\_2 was greater than or equal to SUM\_1 plus the completed New\_Order transactions recorded by the RTE.
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.

All tables are based on 104,040 warehouses, the number of active warehouses during the benchmark was 96,237.

Table Name	Number of Rows
Warehouse	96,237
District	1,040,400
Customer	3,121,200,000
History	3,121,200,000
Order	3,121,200,000
New Order	936,360,000
Order Line	31,211,984,921
Stock	10,404,000,000
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.*

One dual-port FC adapter connected to one DS3400 storage controller was used for the log. The storage controller contained one RAID5 array with 8 disk drives. Each of the disks used for the log had 300GB of storage capacity and the RAID5 LUN was 1952.768GB in size.

All database data was distributed across the 3 SSD 3.5TB and 1 1TB disk. Each SSD was configured as JBOD.

### 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 9.5 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 Warehouse, District, Customer, Order, Order-Line, New Order, History and Stock tables were horizontally partitioned into multiple tables. The specifics of the distribution of partitioned and non-partitioned tables across the physical media can be found in following table:

DATA DISTRIBUTION		
NAME	DATABASE PARTITION	CONTAINER
HDISK0	1,2	D1F01V1ITEM, D1F01V1WARE, D1F01V1DIST, D1F01V1CSTI, D1F01V1NORA, D1F01V1ORL, D1F01V1STK, D1F01V1CST, D1F01V1ORDI, D1F01V1ORD, D1F01V1HIST, D1F01V1NORB, D1F01V2ITEM, D1F01V2WARE, D1F01V2DIST, D1F01V2CSTI, D1F01V2NORA, D1F01V2ORL, D1F01V2STK, D1F01V2CST, D1F01V2ORDI, D1F01V2ORD, D1F01V2HIST, D1F01V2NORB
HDISK1	3,4	D1F01V3ITEM, D1F01V3WARE, D1F01V3DIST, D1F01V3CSTI, D1F01V3NORA, D1F01V3ORL, D1F01V3STK, D1F01V3CST, D1F01V3ORDI, D1F01V3ORD, D1F01V3HIST, D1F01V3NORB, D1F01V4ITEM, D1F01V4WARE, D1F01V4DIST, D1F01V4CSTI, D1F01V4NORA, D1F01V4ORL, D1F01V4STK, D1F01V4CST, D1F01V4ORDI, D1F01V4ORD, D1F01V4HIST, D1F01V4NORB
HDISK2	5,6	D1F01V5ITEM, D1F01V5WARE, D1F01V5DIST, D1F01V5CSTI, D1F01V5NORA, D1F01V5ORL, D1F01V5STK, D1F01V5CST, D1F01V5ORDI, D1F01V5ORD, D1F01V5HIST, D1F01V5NORB, D1F01V6ITEM, D1F01V6WARE, D1F01V6DIST, D1F01V6CSTI, D1F01V6NORA, D1F01V6ORL, D1F01V6STK, D1F01V6CST, D1F01V6ORDI, D1F01V6ORD, D1F01V6HIST, D1F01V6NORB
HDISK3	7,8	D1F01V7ITEM, D1F01V7WARE, D1F01V7DIST, D1F01V7CSTI, D1F01V7NORA, D1F01V7ORL, D1F01V7STK, D1F01V7CST, D1F01V7ORDI, D1F01V7ORD, D1F01V7HIST, D1F01V7NORB, D1F01V8ITEM, D1F01V8WARE, D1F01V8DIST, D1F01V8CSTI, D1F01V8NORA, D1F01V8ORL, D1F01V8STK, D1F01V8CST, D1F01V8ORDI, D1F01V8ORD, D1F01V8HIST, D1F01V8NORB
HDISK4	9,10	D1F01V9ITEM, D1F01V9WARE, D1F01V9DIST, D1F01V9CSTI, D1F01V9NORA, D1F01V9ORL, D1F01V9STK, D1F01V9CST, D1F01V9ORDI, D1F01V9ORD, D1F01V9HIST, D1F01V9NORB, D1F01V10ITEM, D1F01V10WARE, D1F01V10DIST, D1F01V10CSTI, D1F01V10NORA, D1F01V10ORL, D1F01V10STK, D1F01V10CST, D1F01V10ORDI, D1F01V10ORD, D1F01V10HIST, D1F01V10NORB
HDISK5	11,12	D1F01V11ITEM, D1F01V11WARE, D1F01V11DIST, D1F01V11CSTI, D1F01V11NORA, D1F01V11ORL, D1F01V11STK, D1F01V11CST, D1F01V11ORDI, D1F01V11ORD, D1F01V11HIST, D1F01V11NORB, D1F01V12ITEM, D1F01V12WARE, D1F01V12DIST, D1F01V12CSTI, D1F01V12NORA, D1F01V12ORL, D1F01V12STK, D1F01V12CST, D1F01V12ORDI, D1F01V12ORD, D1F01V12HIST, D1F01V12NORB
HDISK6	13,14	D1F01V13ITEM, D1F01V13WARE, D1F01V13DIST, D1F01V13CSTI, D1F01V13NORA, D1F01V13ORL, D1F01V13STK, D1F01V13CST, D1F01V13ORDI, D1F01V13ORD, D1F01V13HIST, D1F01V13NORB, D1F01V14ITEM, D1F01V14WARE, D1F01V14DIST, D1F01V14CSTI, D1F01V14NORA, D1F01V14ORL, D1F01V14STK, D1F01V14CST, D1F01V14ORDI, D1F01V14ORD, D1F01V14HIST, D1F01V14NORB
HDISK7	15,16	D1F01V15ITEM, D1F01V15WARE, D1F01V15DIST, D1F01V15CSTI, D1F01V15NORA, D1F01V15ORL, D1F01V15STK, D1F01V15CST, D1F01V15ORDI, D1F01V15ORD, D1F01V15HIST, D1F01V15NORB, D1F02V1ITEM, D1F02V1WARE, D1F02V1DIST, D1F02V1CSTI, D1F02V1NORA, D1F02V1ORL, D1F02V1STK, D1F02V1CST, D1F02V1ORDI, D1F02V1ORD, D1F02V1HIST, D1F02V1NORB
HDISK8	17,18	D1F02V2ITEM, D1F02V2WARE, D1F02V2DIST, D1F02V2CSTI, D1F02V2NORA, D1F02V2ORL, D1F02V2STK, D1F02V2CST, D1F02V2ORDI, D1F02V2ORD, D1F02V2HIST, D1F02V2NORB, D1F02V3ITEM, D1F02V3WARE, D1F02V3DIST, D1F02V3CSTI, D1F02V3NORA, D1F02V3ORL, D1F02V3STK, D1F02V3CST, D1F02V3ORDI, D1F02V3ORD, D1F02V3HIST, D1F02V3NORB
HDISK9	19,20	D1F02V4ITEM, D1F02V4WARE, D1F02V4DIST, D1F02V4CSTI, D1F02V4NORA, D1F02V4ORL, D1F02V4STK, D1F02V4CST, D1F02V4ORDI, D1F02V4ORD, D1F02V4HIST, D1F02V4NORB, D1F02V5ITEM, D1F02V5WARE, D1F02V5DIST, D1F02V5CSTI, D1F02V5NORA, D1F02V5ORL, D1F02V5STK, D1F02V5CST, D1F02V5ORDI, D1F02V5ORD, D1F02V5HIST, D1F02V5NORB
HDISK10	21,22	D1F02V6ITEM, D1F02V6WARE, D1F02V6DIST, D1F02V6CSTI, D1F02V6NORA, D1F02V6ORL, D1F02V6STK, D1F02V6CST, D1F02V6ORDI, D1F02V6ORD, D1F02V6HIST, D1F02V6NORB, D1F02V7ITEM, D1F02V7WARE, D1F02V7DIST, D1F02V7CSTI, D1F02V7NORA, D1F02V7ORL, D1F02V7STK, D1F02V7CST, D1F02V7ORDI, D1F02V7ORD, D1F02V7HIST, D1F02V7NORB
HDISK11	23,24	D1F02V8ITEM, D1F02V8WARE, D1F02V8DIST, D1F02V8CSTI, D1F02V8NORA, D1F02V8ORL, D1F02V8STK, D1F02V8CST, D1F02V8ORDI, D1F02V8ORD, D1F02V8HIST, D1F02V8NORB, D1F02V9ITEM, D1F02V9WARE, D1F02V9DIST, D1F02V9CSTI, D1F02V9NORA, D1F02V9ORL, D1F02V9STK, D1F02V9CST, D1F02V9ORDI, D1F02V9ORD, D1F02V9HIST, D1F02V9NORB
HDISK12	25,26	D1F02V10ITEM, D1F02V10WARE, D1F02V10DIST, D1F02V10CSTI, D1F02V10NORA, D1F02V10ORL, D1F02V10STK, D1F02V10CST, D1F02V10ORDI, D1F02V10ORD, D1F02V10HIST, D1F02V10NORB, D1F02V11ITEM, D1F02V11WARE, D1F02V11DIST, D1F02V11CSTI, D1F02V11NORA, D1F02V11ORL, D1F02V11STK, D1F02V11CST, D1F02V11ORDI, D1F02V11ORD, D1F02V11HIST, D1F02V11NORB
HDISK13	27,28	D1F02V12ITEM, D1F02V12WARE, D1F02V12DIST, D1F02V12CSTI, D1F02V12NORA, D1F02V12ORL, D1F02V12STK, D1F02V12CST, D1F02V12ORDI, D1F02V12ORD, D1F02V12HIST, D1F02V12NORB, D1F02V13ITEM, D1F02V13WARE, D1F02V13DIST, D1F02V13CSTI, D1F02V13NORA, D1F02V13ORL, D1F02V13STK, D1F02V13CST, D1F02V13ORDI, D1F02V13ORD, D1F02V13HIST, D1F02V13NORB
HDISK14	29,30	D1F02V14ITEM, D1F02V14WARE, D1F02V14DIST, D1F02V14CSTI, D1F02V14NORA, D1F02V14ORL, D1F02V14STK, D1F02V14CST, D1F02V14ORDI, D1F02V14ORD, D1F02V14HIST, D1F02V14NORB, D1F02V15ITEM, D1F02V15WARE, D1F02V15DIST, D1F02V15CSTI, D1F02V15NORA, D1F02V15ORL, D1F02V15STK, D1F02V15CST, D1F02V15ORDI,

		D1F02V15ORD , D1F02V15HIST , D1F02V15NORB
HDISK15	31, 32	D1F03V1ITEM , D1F03V1WARE , D1F03V1DIST , D1F03V1CSTI , D1F03V1NORA , D1F03V1ORL , D1F03V1STK , D1F03V1CST , D1F03V1ORDI , D1F03V1ORD , D1F03V1HIST , D1F03V1NORB , D1F03V2ITEM , D1F03V2WARE , D1F03V2DIST , D1F03V2CSTI , D1F03V2NORA , D1F03V2ORL , D1F03V2STK , D1F03V2CST , D1F03V2ORDI , D1F03V2ORD , D1F03V2HIST , D1F03V2NORB
HDISK16	33, 34	D1F03V3ITEM , D1F03V3WARE , D1F03V3DIST , D1F03V3CSTI , D1F03V3NORA , D1F03V3ORL , D1F03V3STK , D1F03V3CST , D1F03V3ORDI , D1F03V3ORD , D1F03V3HIST , D1F03V3NORB , D1F03V4ITEM , D1F03V4WARE , D1F03V4DIST , D1F03V4CSTI , D1F03V4NORA , D1F03V4ORL , D1F03V4STK , D1F03V4CST , D1F03V4ORDI , D1F03V4ORD , D1F03V4HIST , D1F03V4NORB
HDISK17	35, 36	D1F03V5ITEM , D1F03V5WARE , D1F03V5DIST , D1F03V5CSTI , D1F03V5NORA , D1F03V5ORL , D1F03V5STK , D1F03V5CST , D1F03V5ORDI , D1F03V5ORD , D1F03V5HIST , D1F03V5NORB , D1F03V6ITEM , D1F03V6WARE , D1F03V6DIST , D1F03V6CSTI , D1F03V6NORA , D1F03V6ORL , D1F03V6STK , D1F03V6CST , D1F03V6ORDI , D1F03V6ORD , D1F03V6HIST , D1F03V6NORB
HDISK18	37, 38	D1F03V7ITEM , D1F03V7WARE , D1F03V7DIST , D1F03V7CSTI , D1F03V7NORA , D1F03V7ORL , D1F03V7STK , D1F03V7CST , D1F03V7ORDI , D1F03V7ORD , D1F03V7HIST , D1F03V7NORB , D1F03V8ITEM , D1F03V8WARE , D1F03V8DIST , D1F03V8CSTI , D1F03V8NORA , D1F03V8ORL , D1F03V8STK , D1F03V8CST , D1F03V8ORDI , D1F03V8ORD , D1F03V8HIST , D1F03V8NORB
HDISK19	39, 40	D1F03V9ITEM , D1F03V9WARE , D1F03V9DIST , D1F03V9CSTI , D1F03V9NORA , D1F03V9ORL , D1F03V9STK , D1F03V9CST , D1F03V9ORDI , D1F03V9ORD , D1F03V9HIST , D1F03V9NORB , D1F03V10ITEM , D1F03V10WARE , D1F03V10DIST , D1F03V10CSTI , D1F03V10NORA , D1F03V10ORL , D1F03V10STK , D1F03V10CST , D1F03V10ORDI , D1F03V10ORD , D1F03V10HIST , D1F03V10NORB
HDISK20	42, 42	D1F03V11ITEM , D1F03V11WARE , D1F03V11DIST , D1F03V11CSTI , D1F03V11NORA , D1F03V11ORL , D1F03V11STK , D1F03V11CST , D1F03V11ORDI , D1F03V11ORD , D1F03V11HIST , D1F03V11NORB , D1F03V12ITEM , D1F03V12WARE , D1F03V12DIST , D1F03V12CSTI , D1F03V12NORA , D1F03V12ORL , D1F03V12STK , D1F03V12CST , D1F03V12ORDI , D1F03V12ORD , D1F03V12HIST , D1F03V12NORB
HDISK21	43, 44	D1F03V13ITEM , D1F03V13WARE , D1F03V13DIST , D1F03V13CSTI , D1F03V13NORA , D1F03V13ORL , D1F03V13STK , D1F03V13CST , D1F03V13ORDI , D1F03V13ORD , D1F03V13HIST , D1F03V13NORB , D1F03V14ITEM , D1F03V14WARE , D1F03V14DIST , D1F03V14CSTI , D1F03V14NORA , D1F03V14ORL , D1F03V14STK , D1F03V14CST , D1F03V14ORDI , D1F03V14ORD , D1F03V14HIST , D1F03V14NORB
HDISK22	45, 46	D1F03V15ITEM , D1F03V15WARE , D1F03V15DIST , D1F03V15CSTI , D1F03V15NORA , D1F03V15ORL , D1F03V15STK , D1F03V15CST , D1F03V15ORDI , D1F03V15ORD , D1F03V15HIST , D1F03V15NORB , D1F04V1ITEM , D1F04V1WARE , D1F04V1DIST , D1F04V1CSTI , D1F04V1NORA , D1F04V1ORL , D1F04V1STK , D1F04V1CST , D1F04V1ORDI , D1F04V1ORD , D1F04V1HIST , D1F04V1NORB
HDISK23	47, 48	D1F04V2ITEM , D1F04V2WARE , D1F04V2DIST , D1F04V2CSTI , D1F04V2NORA , D1F04V2ORL , D1F04V2STK , D1F04V2CST , D1F04V2ORDI , D1F04V2ORD , D1F04V2HIST , D1F04V2NORB , D1F04V3ITEM , D1F04V3WARE , D1F04V3DIST , D1F04V3CSTI , D1F04V3NORA , D1F04V3ORL , D1F04V3STK , D1F04V3CST , D1F04V3ORDI , D1F04V3ORD , D1F04V3HIST , D1F04V3NORB
HDISK24	49, 50	D1F04V4ITEM , D1F04V4WARE , D1F04V4DIST , D1F04V4CSTI , D1F04V4NORA , D1F04V4ORL , D1F04V4STK , D1F04V4CST , D1F04V4ORDI , D1F04V4ORD , D1F04V4HIST , D1F04V4NORB , D1F04V5ITEM , D1F04V5WARE , D1F04V5DIST , D1F04V5CSTI , D1F04V5NORA , D1F04V5ORL , D1F04V5STK , D1F04V5CST , D1F04V5ORDI , D1F04V5ORD , D1F04V5HIST , D1F04V5NORB
HDISK25	51, 52	hdisk75 D1F04V6ITEM , D1F04V6WARE , D1F04V6DIST , D1F04V6CSTI , D1F04V6NORA , D1F04V6ORL , D1F04V6STK , D1F04V6CST , D1F04V6ORDI , D1F04V6ORD , D1F04V6HIST , D1F04V6NORB , D1F04V7ITEM , D1F04V7WARE , D1F04V7DIST , D1F04V7CSTI , D1F04V7NORA , D1F04V7ORL , D1F04V7STK , D1F04V7CST , D1F04V7ORDI , D1F04V7ORD , D1F04V7HIST , D1F04V7NORB
HDISK26	53, 54	D1F04V8ITEM , D1F04V8WARE , D1F04V8DIST , D1F04V8CSTI , D1F04V8NORA , D1F04V8ORL , D1F04V8STK , D1F04V8CST , D1F04V8ORDI , D1F04V8ORD , D1F04V8HIST , D1F04V8NORB , D1F04V9ITEM , D1F04V9WARE , D1F04V9DIST , D1F04V9CSTI , D1F04V9NORA , D1F04V9ORL , D1F04V9STK , D1F04V9CST , D1F04V9ORDI , D1F04V9ORD , D1F04V9HIST , D1F04V9NORB
HDISK27	55, 56	D1F04V10ITEM , D1F04V10WARE , D1F04V10DIST , D1F04V10CSTI , D1F04V10NORA , D1F04V10ORL , D1F04V10STK , D1F04V10CST , D1F04V10ORDI , D1F04V10ORD , D1F04V10HIST , D1F04V10NORB , D1F04V11ITEM , D1F04V11WARE , D1F04V11DIST , D1F04V11CSTI , D1F04V11NORA , D1F04V11ORL , D1F04V11STK , D1F04V11CST , D1F04V11ORDI , D1F04V11ORD , D1F04V11HIST , D1F04V11NORB
HDISK28	57, 58	D1F04V12ITEM , D1F04V12WARE , D1F04V12DIST , D1F04V12CSTI , D1F04V12NORA , D1F04V12ORL , D1F04V12STK , D1F04V12CST , D1F04V12ORDI , D1F04V12ORD , D1F04V12HIST , D1F04V12NORB , D1F04V13ITEM , D1F04V13WARE , D1F04V13DIST , D1F04V13CSTI , D1F04V13NORA , D1F04V13ORL , D1F04V13STK , D1F04V13CST , D1F04V13ORDI , D1F04V13ORD , D1F04V13HIST , D1F04V13NORB
HDISK29	59, 60	D1F04V14ITEM , D1F04V14WARE , D1F04V14DIST , D1F04V14CSTI , D1F04V14NORA , D1F04V14ORL , D1F04V14STK , D1F04V14CST , D1F04V14ORDI , D1F04V14ORD , D1F04V14HIST , D1F04V14NORB , D1F04V15ITEM , D1F04V15WARE , D1F04V15DIST , D1F04V15CSTI , D1F04V15NORA , D1F04V15ORL , D1F04V15STK , D1F04V15CST , D1F04V15ORDI , D1F04V15ORD , D1F04V15HIST , D1F04V15NORB

HDISK30	61, 62	D1F05V1ITEM, D1F05V1WARE, D1F05V1DIST, D1F05V1CSTI, D1F05V1NORA, D1F05V1ORL, D1F05V1STK, D1F05V1CST, D1F05V1ORDI, D1F05V1ORD, D1F05V1HIST, D1F05V1NORB, D1F05V2ITEM, D1F05V2WARE, D1F05V2DIST, D1F05V2CSTI, D1F05V2NORA, D1F05V2ORL, D1F05V2STK, D1F05V2CST, D1F05V2ORDI, D1F05V2ORD, D1F05V2HIST, D1F05V2NORB
HDISK31	63, 64	D1F05V3ITEM, D1F05V3WARE, D1F05V3DIST, D1F05V3CSTI, D1F05V3NORA, D1F05V3ORL, D1F05V3STK, D1F05V3CST, D1F05V3ORDI, D1F05V3ORD, D1F05V3HIST, D1F05V3NORB, D1F05V4ITEM, D1F05V4WARE, D1F05V4DIST, D1F05V4CSTI, D1F05V4NORA, D1F05V4ORL, D1F05V4STK, D1F05V4CST, D1F05V4ORDI, D1F05V4ORD, D1F05V4HIST, D1F05V4NORB
HDISK32	65, 66	D1F05V5ITEM, D1F05V5WARE, D1F05V5DIST, D1F05V5CSTI, D1F05V5NORA, D1F05V5ORL, D1F05V5STK, D1F05V5CST, D1F05V5ORDI, D1F05V5ORD, D1F05V5HIST, D1F05V5NORB, D1F05V6ITEM, D1F05V6WARE, D1F05V6DIST, D1F05V6CSTI, D1F05V6NORA, D1F05V6ORL, D1F05V6STK, D1F05V6CST, D1F05V6ORDI, D1F05V6ORD, D1F05V6HIST, D1F05V6NORB
HDISK33	67, 68	D1F05V7ITEM, D1F05V7WARE, D1F05V7DIST, D1F05V7CSTI, D1F05V7NORA, D1F05V7ORL, D1F05V7STK, D1F05V7CST, D1F05V7ORDI, D1F05V7ORD, D1F05V7HIST, D1F05V7NORB, D1F05V8ITEM, D1F05V8WARE, D1F05V8DIST, D1F05V8CSTI, D1F05V8NORA, D1F05V8ORL, D1F05V8STK, D1F05V8CST, D1F05V8ORDI, D1F05V8ORD, D1F05V8HIST, D1F05V8NORB
HDISK34	69, 70	D1F05V9ITEM, D1F05V9WARE, D1F05V9DIST, D1F05V9CSTI, D1F05V9NORA, D1F05V9ORL, D1F05V9STK, D1F05V9CST, D1F05V9ORDI, D1F05V9ORD, D1F05V9HIST, D1F05V9NORB, D1F05V10ITEM, D1F05V10WARE, D1F05V10DIST, D1F05V10CSTI, D1F05V10NORA, D1F05V10ORL, D1F05V10STK, D1F05V10CST, D1F05V10ORDI, D1F05V10ORD, D1F05V10HIST, D1F05V10NORB
HDISK35	71, 72	D1F05V11ITEM, D1F05V11WARE, D1F05V11DIST, D1F05V11CSTI, D1F05V11NORA, D1F05V11ORL, D1F05V11STK, D1F05V11CST, D1F05V11ORDI, D1F05V11ORD, D1F05V11HIST, D1F05V11NORB, D1F05V12ITEM, D1F05V12WARE, D1F05V12DIST, D1F05V12CSTI, D1F05V12NORA, D1F05V12ORL, D1F05V12STK, D1F05V12CST, D1F05V12ORDI, D1F05V12ORD, D1F05V12HIST, D1F05V12NORB
HDISK36	73, 74	D1F05V13ITEM, D1F05V13WARE, D1F05V13DIST, D1F05V13CSTI, D1F05V13NORA, D1F05V13ORL, D1F05V13STK, D1F05V13CST, D1F05V13ORDI, D1F05V13ORD, D1F05V13HIST, D1F05V13NORB, D1F05V14ITEM, D1F05V14WARE, D1F05V14DIST, D1F05V14CSTI, D1F05V14NORA, D1F05V14ORL, D1F05V14STK, D1F05V14CST, D1F05V14ORDI, D1F05V14ORD, D1F05V14HIST, D1F05V14NORB
HDISK37	75, 76	D1F05V15ITEM, D1F05V15WARE, D1F05V15DIST, D1F05V15CSTI, D1F05V15NORA, D1F05V15ORL, D1F05V15STK, D1F05V15CST, D1F05V15ORDI, D1F05V15ORD, D1F05V15HIST, D1F05V15NORB, D1F06V1ITEM, D1F06V1WARE, D1F06V1DIST, D1F06V1CSTI, D1F06V1NORA, D1F06V1ORL, D1F06V1STK, D1F06V1CST, D1F06V1ORDI, D1F06V1ORD, D1F06V1HIST, D1F06V1NORB
HDISK38	77, 78	D1F06V2ITEM, D1F06V2WARE, D1F06V2DIST, D1F06V2CSTI, D1F06V2NORA, D1F06V2ORL, D1F06V2STK, D1F06V2CST, D1F06V2ORDI, D1F06V2ORD, D1F06V2HIST, D1F06V2NORB, D1F06V3ITEM, D1F06V3WARE, D1F06V3DIST, D1F06V3CSTI, D1F06V3NORA, D1F06V3ORL, D1F06V3STK, D1F06V3CST, D1F06V3ORDI, D1F06V3ORD, D1F06V3HIST, D1F06V3NORB
HIDSK39	79, 80	D1F06V4ITEM, D1F06V4WARE, D1F06V4DIST, D1F06V4CSTI, D1F06V4NORA, D1F06V4ORL, D1F06V4STK, D1F06V4CST, D1F06V4ORDI, D1F06V4ORD, D1F06V4HIST, D1F06V4NORB, D1F06V5ITEM, D1F06V5WARE, D1F06V5DIST, D1F06V5CSTI, D1F06V5NORA, D1F06V5ORL, D1F06V5STK, D1F06V5CST, D1F06V5ORDI, D1F06V5ORD, D1F06V5HIST, D1F06V5NORB
HDISK40	81, 82	D1F06V6ITEM, D1F06V6WARE, D1F06V6DIST, D1F06V6CSTI, D1F06V6NORA, D1F06V6ORL, D1F06V6STK, D1F06V6CST, D1F06V6ORDI, D1F06V6ORD, D1F06V6HIST, D1F06V6NORB, D1F06V7ITEM, D1F06V7WARE, D1F06V7DIST, D1F06V7CSTI, D1F06V7NORA, D1F06V7ORL, D1F06V7STK, D1F06V7CST, D1F06V7ORDI, D1F06V7ORD, D1F06V7HIST, D1F06V7NORB
HDISK41	83, 84	D1F06V8ITEM, D1F06V8WARE, D1F06V8DIST, D1F06V8CSTI, D1F06V8NORA, D1F06V8ORL, D1F06V8STK, D1F06V8CST, D1F06V8ORDI, D1F06V8ORD, D1F06V8HIST, D1F06V8NORB, D1F06V9ITEM, D1F06V9WARE, D1F06V9DIST, D1F06V9CSTI, D1F06V9NORA, D1F06V9ORL, D1F06V9STK, D1F06V9CST, D1F06V9ORDI, D1F06V9ORD, D1F06V9HIST, D1F06V9NORB
HDISK42	85, 86	D1F06V10ITEM, D1F06V10WARE, D1F06V10DIST, D1F06V10CSTI, D1F06V10NORA, D1F06V10ORL, D1F06V10STK, D1F06V10CST, D1F06V10ORDI, D1F06V10ORD, D1F06V10HIST, D1F06V10NORB, D1F06V11ITEM, D1F06V11WARE, D1F06V11DIST, D1F06V11CSTI, D1F06V11NORA, D1F06V11ORL, D1F06V11STK, D1F06V11CST, D1F06V11ORDI, D1F06V11ORD, D1F06V11HIST, D1F06V11NORB
HDISK43	87, 88	D1F06V12ITEM, D1F06V12WARE, D1F06V12DIST, D1F06V12CSTI, D1F06V12NORA, D1F06V12ORL, D1F06V12STK, D1F06V12CST, D1F06V12ORDI, D1F06V12ORD, D1F06V12HIST, D1F06V12NORB, D1F06V13ITEM, D1F06V13WARE, D1F06V13DIST, D1F06V13CSTI, D1F06V13NORA, D1F06V13ORL, D1F06V13STK, D1F06V13CST, D1F06V13ORDI, D1F06V13ORD, D1F06V13HIST, D1F06V13NORB
HDISK44	89, 90	D1F06V14ITEM, D1F06V14WARE, D1F06V14DIST, D1F06V14CSTI, D1F06V14NORA, D1F06V14ORL, D1F06V14STK, D1F06V14CST, D1F06V14ORDI, D1F06V14ORD, D1F06V14HIST, D1F06V14NORB, D1F06V15ITEM, D1F06V15WARE, D1F06V15DIST, D1F06V15CSTI, D1F06V15NORA, D1F06V15ORL, D1F06V15STK, D1F06V15CST, D1F06V15ORDI, D1F06V15ORD, D1F06V15HIST, D1F06V15NORB
HDISK45	91, 92	D1F07V1ITEM, D1F07V1WARE, D1F07V1DIST, D1F07V1CSTI, D1F07V1NORA, D1F07V1ORL, D1F07V1STK, D1F07V1CST, D1F07V1ORDI, D1F07V1ORD, D1F07V1HIST, D1F07V1NORB, D1F07V2ITEM, D1F07V2WARE, D1F07V2DIST, D1F07V2CSTI, D1F07V2NORA, D1F07V2ORL, D1F07V2STK, D1F07V2CST, D1F07V2ORDI, D1F07V2ORD, D1F07V2HIST, D1F07V2NORB

HDISK46	93, 94	D1F07V3ITEM, D1F07V3WARE, D1F07V3DIST, D1F07V3CSTI, D1F07V3NORA, D1F07V3ORL, D1F07V3STK, D1F07V3CST, D1F07V3ORDI, D1F07V3ORD, D1F07V3HIST, D1F07V3NORB, D1F07V4ITEM, D1F07V4WARE, D1F07V4DIST, D1F07V4CSTI, D1F07V4NORA, D1F07V4ORL, D1F07V4STK, D1F07V4CST, D1F07V4ORDI, D1F07V4ORD, D1F07V4HIST, D1F07V4NORB
HDISK47	95, 96	D1F07V5ITEM, D1F07V5WARE, D1F07V5DIST, D1F07V5CSTI, D1F07V5NORA, D1F07V5ORL, D1F07V5STK, D1F07V5CST, D1F07V5ORDI, D1F07V5ORD, D1F07V5HIST, D1F07V5NORB, D1F07V6ITEM, D1F07V6WARE, D1F07V6DIST, D1F07V6CSTI, D1F07V6NORA, D1F07V6ORL, D1F07V6STK, D1F07V6CST, D1F07V6ORDI, D1F07V6ORD, D1F07V6HIST, D1F07V6NORB
HDISK48	97, 98	D1F07V7ITEM, D1F07V7WARE, D1F07V7DIST, D1F07V7CSTI, D1F07V7NORA, D1F07V7ORL, D1F07V7STK, D1F07V7CST, D1F07V7ORDI, D1F07V7ORD, D1F07V7HIST, D1F07V7NORB, D1F07V8ITEM, D1F07V8WARE, D1F07V8DIST, D1F07V8CSTI, D1F07V8NORA, D1F07V8ORL, D1F07V8STK, D1F07V8CST, D1F07V8ORDI, D1F07V8ORD, D1F07V8HIST, D1F07V8NORB
HDISK49	99, 100	D1F07V9ITEM, D1F07V9WARE, D1F07V9DIST, D1F07V9CSTI, D1F07V9NORA, D1F07V9ORL, D1F07V9STK, D1F07V9CST, D1F07V9ORDI, D1F07V9ORD, D1F07V9HIST, D1F07V9NORB, D1F07V10ITEM, D1F07V10WARE, D1F07V10DIST, D1F07V10CSTI, D1F07V10NORA, D1F07V10ORL, D1F07V10STK, D1F07V10CST, D1F07V10ORDI, D1F07V10ORD, D1F07V10HIST, D1F07V10NORB
HDISK50	101, 102	D1F07V11ITEM, D1F07V11WARE, D1F07V11DIST, D1F07V11CSTI, D1F07V11NORA, D1F07V11ORL, D1F07V11STK, D1F07V11CST, D1F07V11ORDI, D1F07V11ORD, D1F07V11HIST, D1F07V11NORB, D1F07V12ITEM, D1F07V12WARE, D1F07V12DIST, D1F07V12CSTI, D1F07V12NORA, D1F07V12ORL, D1F07V12STK, D1F07V12CST, D1F07V12ORDI, D1F07V12ORD, D1F07V12HIST, D1F07V12NORB
HDISK51	103, 104	D1F07V13ITEM, D1F07V13WARE, D1F07V13DIST, D1F07V13CSTI, D1F07V13NORA, D1F07V13ORL, D1F07V13STK, D1F07V13CST, D1F07V13ORDI, D1F07V13ORD, D1F07V13HIST, D1F07V13NORB, D1F07V14ITEM, D1F07V14WARE, D1F07V14DIST, D1F07V14CSTI, D1F07V14NORA, D1F07V14ORL, D1F07V14STK, D1F07V14CST, D1F07V14ORDI, D1F07V14ORD, D1F07V14HIST, D1F07V14NORB
HDISK52	105, 106	D1F07V15ITEM, D1F07V15WARE, D1F07V15DIST, D1F07V15CSTI, D1F07V15NORA, D1F07V15ORL, D1F07V15STK, D1F07V15CST, D1F07V15ORDI, D1F07V15ORD, D1F07V15HIST, D1F07V15NORB, D1F08V1ITEM, D1F08V1WARE, D1F08V1DIST, D1F08V1CSTI, D1F08V1NORA, D1F08V1ORL, D1F08V1STK, D1F08V1CST, D1F08V1ORDI, D1F08V1ORD, D1F08V1HIST, D1F08V1NORB
HDISK53	107, 108	D1F08V2ITEM, D1F08V2WARE, D1F08V2DIST, D1F08V2CSTI, D1F08V2NORA, D1F08V2ORL, D1F08V2STK, D1F08V2CST, D1F08V2ORDI, D1F08V2ORD, D1F08V2HIST, D1F08V2NORB, D1F08V3ITEM, D1F08V3WARE, D1F08V3DIST, D1F08V3CSTI, D1F08V3NORA, D1F08V3ORL, D1F08V3STK, D1F08V3CST, D1F08V3ORDI, D1F08V3ORD, D1F08V3HIST, D1F08V3NORB
HDISK54	109, 110	D1F08V4ITEM, D1F08V4WARE, D1F08V4DIST, D1F08V4CSTI, D1F08V4NORA, D1F08V4ORL, D1F08V4STK, D1F08V4CST, D1F08V4ORDI, D1F08V4ORD, D1F08V4HIST, D1F08V4NORB, D1F08V5ITEM, D1F08V5WARE, D1F08V5DIST, D1F08V5CSTI, D1F08V5NORA, D1F08V5ORL, D1F08V5STK, D1F08V5CST, D1F08V5ORDI, D1F08V5ORD, D1F08V5HIST, D1F08V5NORB
HDISK55	111, 112	D1F08V6ITEM, D1F08V6WARE, D1F08V6DIST, D1F08V6CSTI, D1F08V6NORA, D1F08V6ORL, D1F08V6STK, D1F08V6CST, D1F08V6ORDI, D1F08V6ORD, D1F08V6HIST, D1F08V6NORB, D1F08V7ITEM, D1F08V7WARE, D1F08V7DIST, D1F08V7CSTI, D1F08V7NORA, D1F08V7ORL, D1F08V7STK, D1F08V7CST, D1F08V7ORDI, D1F08V7ORD, D1F08V7HIST, D1F08V7NORB
HDISK56	113-120	D1F08V8ITEM, D1F08V8WARE, D1F08V8DIST, D1F08V8CSTI, D1F08V8NORA, D1F08V8ORL, D1F08V8STK, D1F08V8CST, D1F08V8ORDI, D1F08V8ORD, D1F08V8HIST, D1F08V8NORB, D1F08V9ITEM, D1F08V9WARE, D1F08V9DIST, D1F08V9CSTI, D1F08V9NORA, D1F08V9ORL, D1F08V9STK, D1F08V9CST, D1F08V9ORDI, D1F08V9ORD, D1F08V9HIST, D1F08V9NORB, D1F08V10ITEM, D1F08V10WARE, D1F08V10DIST, D1F08V10CSTI, D1F08V10NORA, D1F08V10ORL, D1F08V10STK, D1F08V10CST, D1F08V10ORDI, D1F08V10ORD, D1F08V10HIST, D1F08V10NORB, D1F08V11ITEM, D1F08V11WARE, D1F08V11DIST, D1F08V11CSTI, D1F08V11NORA, D1F08V11ORL, D1F08V11STK, D1F08V11CST, D1F08V11ORDI, D1F08V11ORD, D1F08V11HIST, D1F08V11NORB, D1F08V12ITEM, D1F08V12WARE, D1F08V12DIST, D1F08V12CSTI, D1F08V12NORA, D1F08V12ORL, D1F08V12STK, D1F08V12CST, D1F08V12ORDI, D1F08V12ORD, D1F08V12HIST, D1F08V12NORB, D1F08V13ITEM, D1F08V13WARE, D1F08V13DIST, D1F08V13CSTI, D1F08V13NORA, D1F08V13ORL, D1F08V13STK, D1F08V13CST, D1F08V13ORDI, D1F08V13ORD, D1F08V13HIST, D1F08V13NORB, D1F08V14ITEM, D1F08V14WARE, D1F08V14DIST, D1F08V14CSTI, D1F08V14NORA, D1F08V14ORL, D1F08V14STK, D1F08V14CST, D1F08V14ORDI, D1F08V14ORD, D1F08V14HIST, D1F08V14NORB, D1F08V15ITEM, D1F08V15WARE, D1F08V15DIST, D1F08V15CSTI, D1F08V15NORA, D1F08V15ORL, D1F08V15STK, D1F08V15CST, D1F08V15ORDI, D1F08V15ORD, D1F08V15HIST, D1F08V15NORB

**Table 4-2:** IBM Power 780 Server Model 9179-MHB 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	104,040					
Measured TpmC	1,200,011					
<b>Table</b>	<b>Rows</b>	<b>Table</b>	<b>Index</b>	<b>5% Space</b>	<b>Total Space</b>	
Warehouse	104,040		75	0	4	79
District	1,040,400		240	0	12	252
Item	100,000		10	0	1	11
Stock	10,404,000,000	3,387,240		0	169,362	3,556,602
Customer	3,121,200,000	2,439,000		151,200	129,510	2,719,710
New-Order	936,360,000	77,760		0	3,888	81,648
Orders	3,121,200,000	120,372		88,080	0	208,452
Order-Line	31,212,000,000	2,093,748		0	0	2,093,748
History	3,121,200,000	194,400		0	0	194,400
Additional Overhead		1,927,560				1,927,560
Free Space	334,924					
Dynamic Space	2,408,520					
Static Space	8,373,941					
Daily Growth	444,483					
Daily Spread	0					
				<u>30 Minute log Computations</u>		
					Log Written (KB)	81,000,743
					New-Order Txns	36,000,330
					Log Written per New-Order (KB)	2.25
<b>Data Storage Requirement</b>						
60 Days (MB)	35,042,919					
60 Days (GB)	34,222					
<b>Log Storage Requirement</b>						
8 Hours (GB)	1,235.97					
<b>Disk Sizing</b>						
	<b>Formatted</b>		<b>SUT</b>		<b>Priced</b>	
<b>Disk Type</b>	<b>Capacity (GB)</b>	<b># of Disks</b>	<b>Capacity (GB)</b>	<b># of Disks</b>	<b>Capacity (GB)</b>	
DB 177 GB	186.30	56	10,433	60	11,178	
DB 1 TB	931.00	1	931	26	24,206	
LOG 300 GB	244.00	8	1,952	8	1,952	
OS 146 GB	136.50	4	546	2	273	
<b>Total Capacity</b>					37,609	

## 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
<b>90 %</b>	0.694	0.689	0.684	0.459/0.48	0.719	0.464
<b>Average</b>	0.403	0.3970	0.395	0.234/0.26	0.421	0.236
<b>Maximum</b>	12.522	12.229	9.244	8.912/10.42	9.801	12.411
Think Times						
<b>Minimum</b>	0.01	0.01	0.01	0.01	0.01	N/A
<b>Average</b>	12.02	12.02	10.01	5.02	5.02	N/A
<b>Maximum</b>	120.21	120.21	100.10	50.20	50.21	N/A
Keying Times						
<b>Minimum</b>	18.01	3.01	2.01	2.01	2.01	N/A
<b>Average</b>	18.01	3.01	2.01	2.01	2.01	N/A
<b>Maximum</b>	18.03	3.03	2.03	2.03	2.03	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.

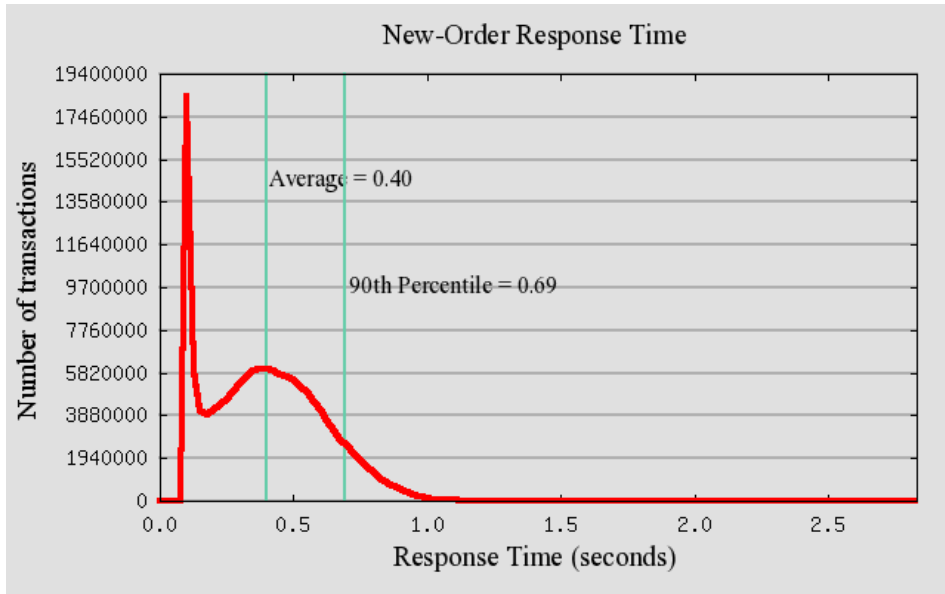


Figure 5-1: New-Order Response Time Distribution

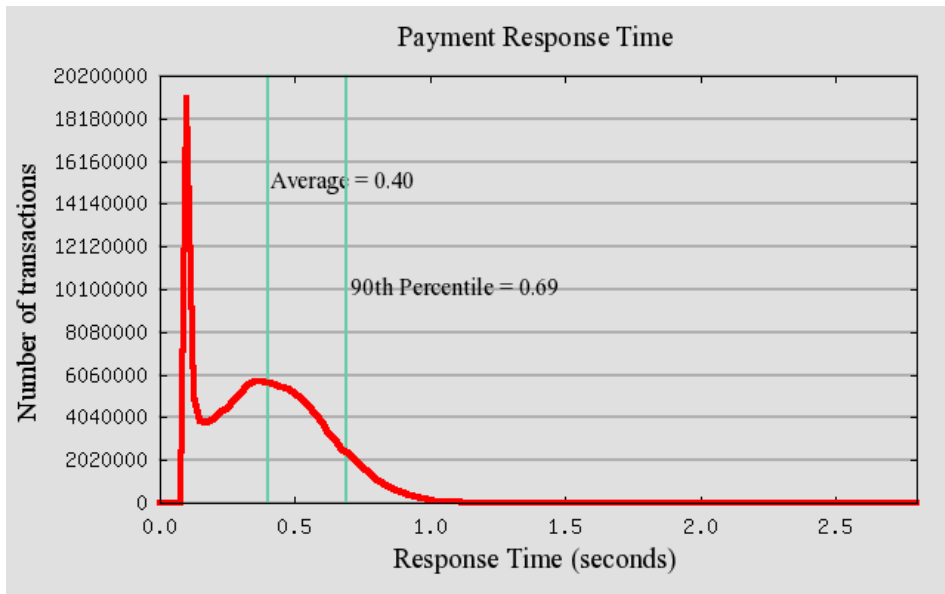
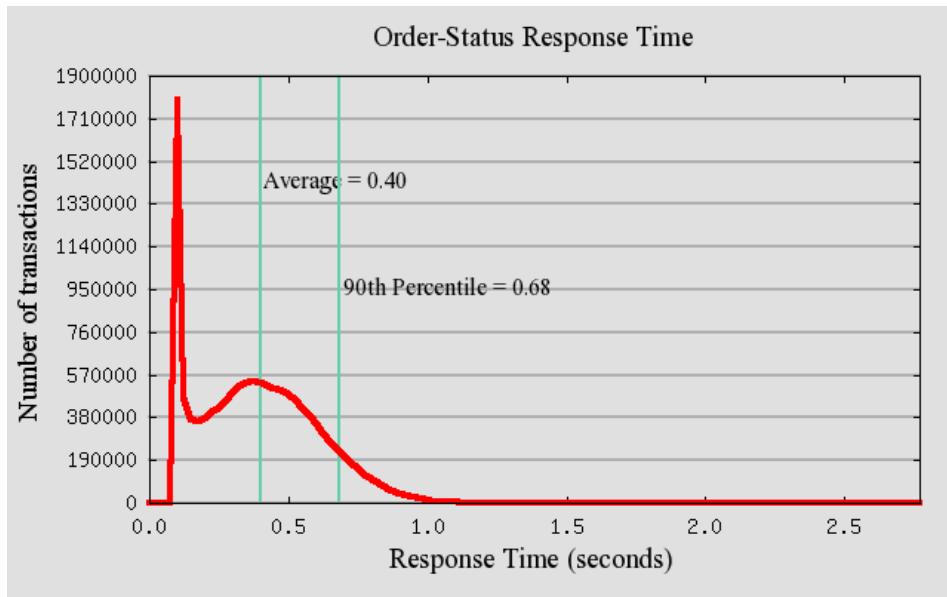
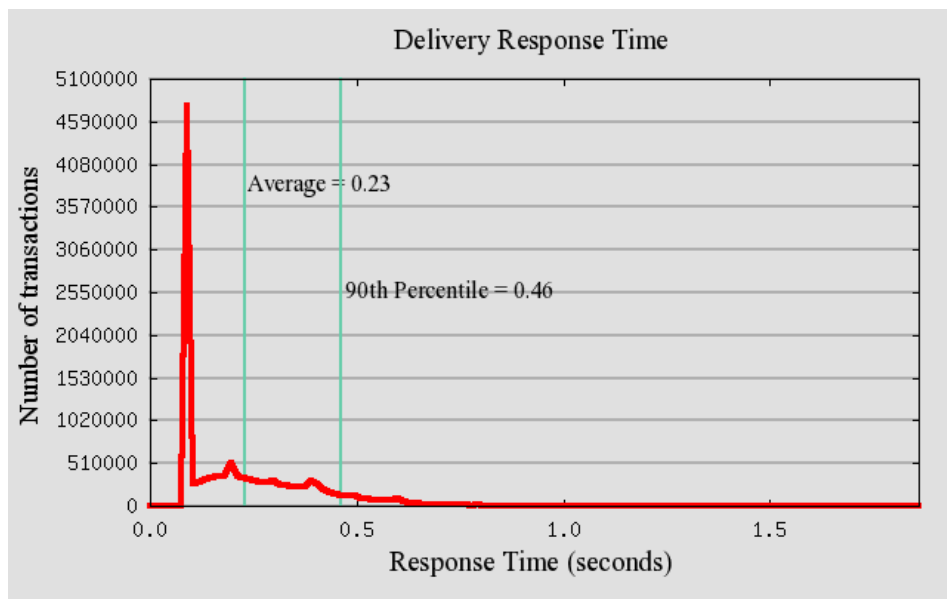


Figure 5-2: Payment Response Time Distribution



**Figure 5-3: Order-Status Response Time Distribution**



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

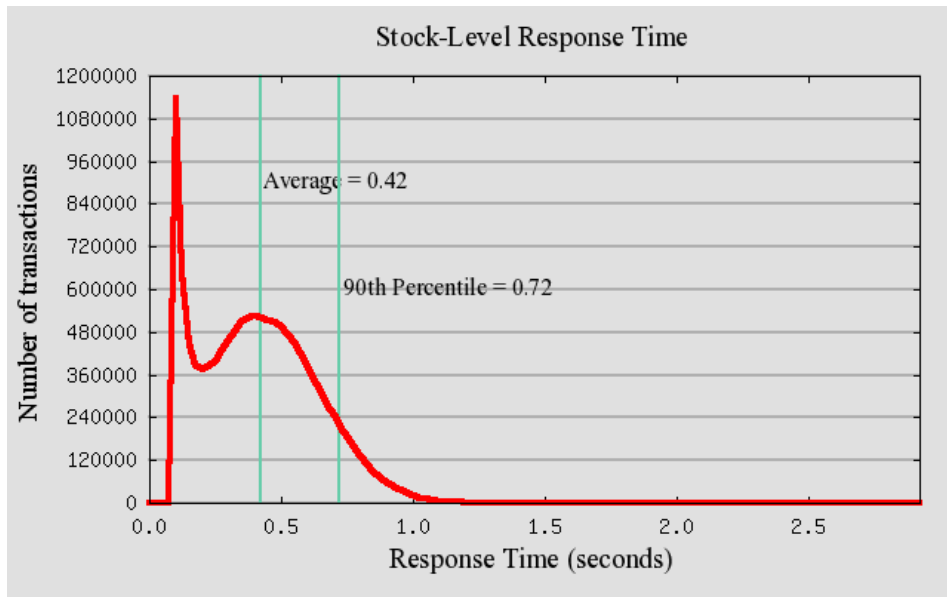


Figure 5-5: Stock Level Response Time Distribution

#### 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.

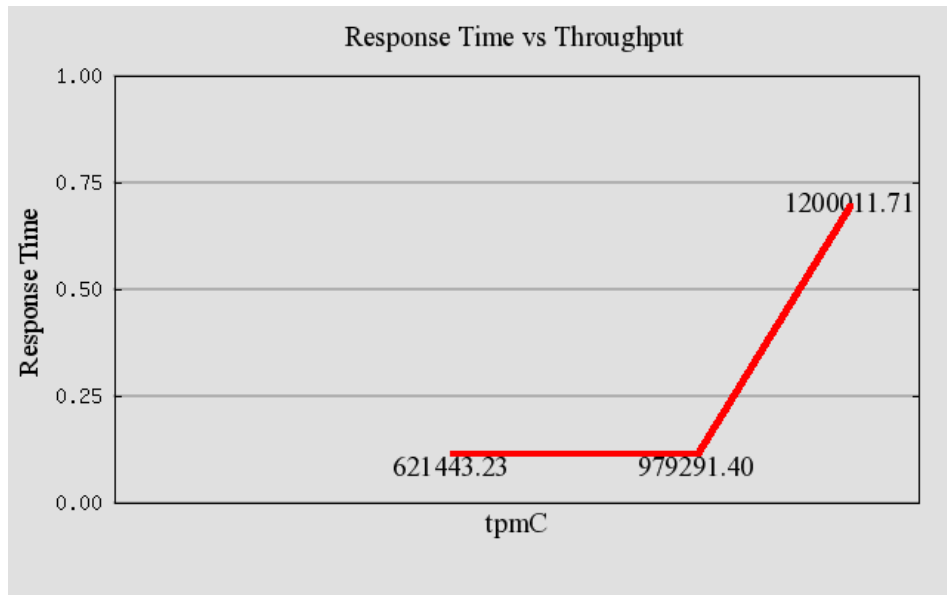


Figure 5-6: New-Order Response Time vs. Throughput

## 5.5. Think Time Frequency Distribution

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

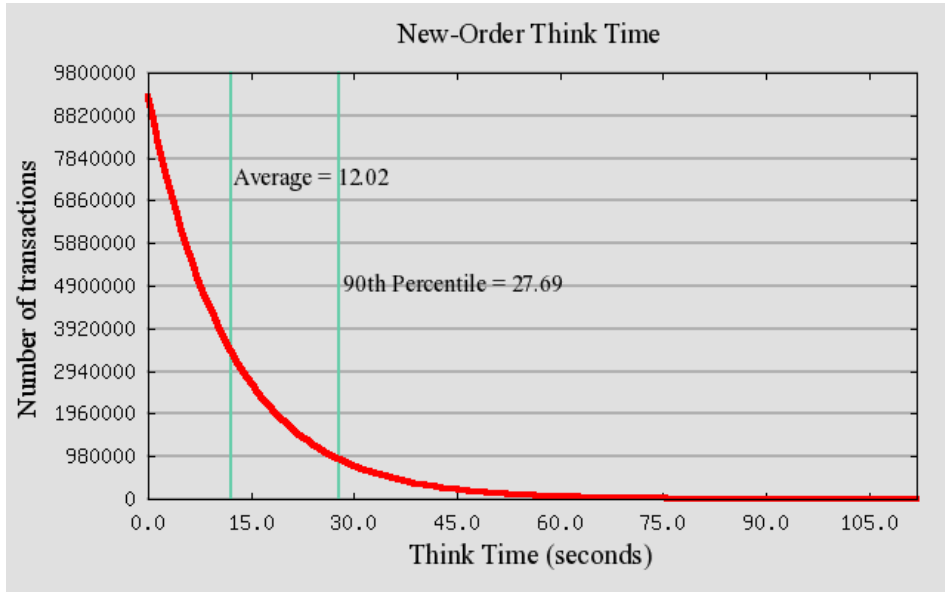


Figure 5-7: New-Order Think Time Distribution

## 5.6. Throughput versus Elapsed Time

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

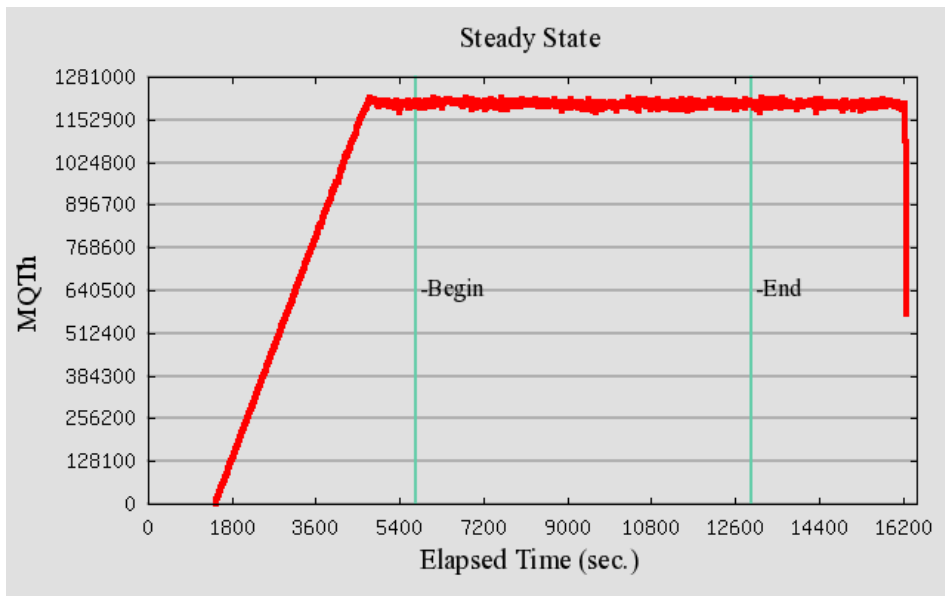


Figure 5-8: New-Order Throughput vs. Elapsed Time

## 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 0-minute measurement interval was used to guarantee that all work normally performed during an 8-hour sustained test are 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 5.1 (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 5.1(IIS). 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 2008 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 9.5 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 9.5 proceeds to update the database as follows:

When DB2 9.5 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 9.5 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 9.5 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 9.5. For a more detailed description of the general principles of the write-ahead-logging protocol, see the IBM research paper, “ARIES: A Transaction Recovery Method Supporting Fine Granularity Locking and Partial Rollbacks Using Write-Ahead Logging,” by C. Mohan, Database Technology Institute, IBM Almaden Research Center.

(<http://portal.acm.org/citation.cfm?id=128770&coll=portal&dl=ACM&CFID=10343790&CFTOKEN=42047146>)

## 5.9. Measurement Interval

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

A 2-hour 0-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. Appendix D contains the scripts used in the testing.

### **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 database system was connected to 1 Ethernet Gigabit switch with a rate of 1000Mbits 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 software for the priced configuration is listed in the pricing sheets as part of the executive summary. Third Party Pricing Information is provided in Appendix - D:.

### 7.2. Three Year Cost of System Configuration

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

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

A 45.7% discount was based on the overall value of the specific components from IBM in the quotation provided in Appendix - D:.. Discounts for similarly sized configurations with similar quantities and configurations will be similar to those quoted here.

### 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 or before: October 13, 2010

### 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 Power 780 Model 9179-MHB	1,200,011	\$825,004 USD	\$0.69 USD	October 13, 2010

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

### 7.5. Country-specific pricing

*Additional Clause 7 related items may be included in the Full Disclosure Report for each country specific priced configuration. Country specific pricing is subject to Clause 7.1.7*

This system is being priced for the United States of America. All prices are based on IBM US list prices.

### 7.6. Orderability Date

*For each of the components that are not orderable on the report date of the FDR, the following information must be included in the FDR:*

- *Name and part number of the item that is not orderable*
- *The date when the component can be ordered (on or before the Availability Date)*

- *The method to be used to order the component (at or below the quoted price) when that date arrives*
- *The method for verifying the price*

- 0/128GB(4X32GB) SDRAM DDR3 DIMMS, 1066MHZ , part number 5602, availability date October 13, 2010. This item can be ordered through IBM see Appendix D.
- 3.5TB SSD Package, part number FC4367, availability date October 13, 2010. This item can be ordered through IBM see Appendix D.

Prices for all items used in this benchmark can be verified through the contact information provided in the pricing quote for the appropriate vendor. Price quotes are included in Appendix - D:

---

## **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:

William Bostic  
 IBM Power System Performance  
 11501 Burnet Road  
 Austin, TX 78758

Berni Schiefer  
 IBM Information Management Performance  
 8200 Warden Avenue  
 Markham, Ontario L6G1C7

April 7, 2010

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

Platform: IBM Power 780 Model 9179-MHB  
 Operating system: AIX V6.1  
 Database Manager: DB2 9.5  
 Transaction Manager: Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	New Order 90% Response Time	tpmC
<b>Server: IBM Power 780 Model 9179-MHB</b>				
2 processors with 4 Turbocore POWER7 (4.14GHz)	512 GB (8 x 4MB L3, 16MB shared cache)	2 x 146GB 15K rpm SAS (int.) 8 x 300GB 15K rpm 26 x 1TB 7.2K rpm SATA 3 x 3.5TB SSD Package	0.694 Seconds	1,200,011
<b>8 Clients: IBM System x3550 M2 (each with)</b>				
1 x Intel Xeon Quad-core (2.4 GHz)	3 GB (8 MB L3)	1 x 146GB 15K rpm 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 120 minutes
- Write-ahead-logging was active during the measurement interval
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

Respectfully Yours,

A handwritten signature in black ink, appearing to read "François Raab". The signature is fluid and cursive, written in a professional style.

François Raab, President

# Appendix - A: Client Server Code

## A.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/Winx64 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 -DSWAP_ENDIAN
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=$(LD_FLAGS_SHLIB) /DEF:rpctpc.def
LD_FLAGS_LIB=LIBPATH:$(TPCC_SQLLIB)\lib /LIBPATH:"C:\MSDKx64\lib\amd64"
/LIBPATH:"C:\Program Files\Microsoft SDKs\Windows\v6.0A\lib" /LIBPATH:"C:\Program Files
(x86)\Microsoft Visual Studio 9.0\VC\lib" winmm.lib db2api.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
# #####

PRP_OPTS = PACKAGE \
           ISOLATION RR \
           QUERYOPT 7 \
           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)

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

LIBS = tpcccli$(LIBEXT)

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

all: connect $(OBJ) plan $(LIBS) disconnect
     $(AR) $(ARFLAGS) $(ARFLAGS_OUT)tpcccli$(LIBEXT) $(OBJ) $(ARFLAGS_LIB)
     @echo "-----"
     @echo "Please copy lval.h, db2tpcc.h, and tpcccli$(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

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

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

tpcccli.c:
     @echo "Prepping $*.sqc"
     db2 prep $*.sqc $(PRP_OPTS)
     db2 grant execute on package TPCCCLI to public

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

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

# Source
tpcccli$(OBJEXT): tpcccli.c

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

### Src.Cli/client-setup.pl

```
#!/bin/perl

#####
## 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.
#####

# client-setup.pl - Setups tpcccli.sqc for client/server varying schemas
#
#
use lib $ENV{TPCC_ROOT};
use tpccutil;
use strict;
use File::Copy;

my $SERVER_SLASH = "";
checkenv('TPCC_ROOT', 'SLASH', 'PLATFORM', 'TPCC_SCHEMA', 'SERVER_TPCC_SCHEMA');

if ($myEnv{ENV_TPCC_SCHEMA}!=$myEnv{ENV_SERVER_TPCC_SCHEMA}){

    my $cli_file = join ($myEnv{ENV_SLASH}, $myEnv{ENV_TPCC_ROOT}, 'Src.Cli', 'tpcccli.sqc');
    my $cli_backup = join ($myEnv{ENV_SLASH}, $myEnv{ENV_TPCC_ROOT}, 'Src.Cli',
'tpcccli.sqc.backup');
    move($cli_file, $cli_backup) || die "Could not move $cli_file to $cli_backup\n";
    open (fn_cli, $cli_backup) || die "Could not find $cli_file\n";
```

```

open (fh_cli,">$cli_file") || die "Could not create file : $cli_file\n";
while (<fh_cli>{

    /EXECIS+SQLIS+CALLIS+news/i and do {
        s/news/$myEnv{ENV_SERVER_TPCC_SCHEMA}.news/;
    };
    /EXECIS+SQLIS+CALLIS+ords/i and do {
        s/ords/$myEnv{ENV_SERVER_TPCC_SCHEMA}.ords/;
    };
    /EXECIS+SQLIS+CALLIS+dels/i and do {
        s/dels/$myEnv{ENV_SERVER_TPCC_SCHEMA}.dels/;
    };
    /FROMIS+TABLEIS+(\s*"PAY_C_LAST"/ and do {
        s/PAY_C_LAST/$myEnv{ENV_SERVER_TPCC_SCHEMA}.PAY_C_LAST/;
    };
    /FROMIS+TABLEIS+(\s*"PAY_C_ID"/ and do {
        s/PAY_C_ID/$myEnv{ENV_SERVER_TPCC_SCHEMA}.PAY_C_ID/;
    };
    /FROMIS+ORDER_LINEIS+(\s*"STOCKIS"/ and do {
        s/ORDER_LINE/$myEnv{ENV_SERVER_TPCC_SCHEMA}.ORDER_LINE/;
        s/STO/$myEnv{ENV_SERVER_TPCC_SCHEMA}.STO/;
        s/DISTRIS/$myEnv{ENV_SERVER_TPCC_SCHEMA}.DISTR/;
    };

};

print fh_cli $;

}

close (fh_cli);
close (fh_cli);
}

```

## Src.Cli/tpcccli.sqc

```

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

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

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

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

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

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

```

```

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

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

    EXEC SQL BEGIN DECLARE SECTION;

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

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

    EXEC SQL END DECLARE SECTION;

    int clientRc = TRAN_OK ;

    int itemIndex = 0 ;

    in_neword->s_all_local = 1 ;
    for ( itemIndex = 0 ;
          itemIndex < 15 && in_neword->in_item[ itemIndex ].s_OL_I_ID != UNUSED_ITEM_ID ;
          itemIndex++ )
    {
        if ( in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID != in_neword->s_W_ID )
        {
            in_neword->s_all_local = 0 ;
        }
    }

    in_neword->s_O_OL_CNT = itemIndex ;

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

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

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

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

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

```

```

    SWAP_BYTE(in_neword->s_D_ID);
    SWAP_BYTE(in_neword->s_O_OL_CNT);
    SWAP_BYTE(in_neword->s_all_local);
    SWAP_BYTE(in_neword->duplicate_items);
#endif //SWAP_ENDIAN

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

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

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

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

        // Compute order total

        neword->s_total_amount = 0 ;

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

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

        }
    }

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

```



```

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

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

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

return ( clientRc );
}

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

int payment_sql ( struct in_payment_struct * in_payment
                , struct out_payment_struct * payment )
{
    struct sqlca sqlca;

    int clientRc = TRAN_OK;

    EXEC SQL BEGIN DECLARE SECTION;

    // Inputs

    float h_amount;
    sqlint32 in_c_id;

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

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

    // Outputs

    sqlint32 c_id;

    double c_credit_lim;
    float c_discount;
    double c_balance;

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

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

    char c_last[ 16 ];

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

    char c_credit[ 2 ];

    char c_since[ 27 ];

    char c_data[ 200 ];
    short c_data_indicator = 0;

    char h_date[ 27 ];

```

```

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

    EXEC SQL END DECLARE SECTION;

    // Input redirects

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

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

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

    // Output redirects

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

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

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

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

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

    #define h_date        payment->s_H_DATE_time

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

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

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

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

    if ( in_c_id == 0 )
    {
        c_data_prefix_c_last.len = sprintf( c_data_prefix_c_last.data, "%2.2d%6.6d%2.2d%6.6d%06.2f",
        c_d_id, c_w_id, d_id, w_id, h_amount );

        // Setup the input c_last varchar

```

```

        c_last_input.len = strlen( in_payment->s_C_LAST );
        memcpy( c_last_input.data, in_payment->s_C_LAST, c_last_input.len );

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

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

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

    }

    retry_tran:

    payment->deadlocks ++;

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

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

        INTO :w_street_1, :w_street_2, :w_city, :w_state, :w_zip
            , :d_street_1, :d_street_2, :d_city, :d_state, :d_zip
            , :c_id, :c_first, :c_middle, :c_street_1, :c_street_2, :c_city, :c_state
            , :c_zip, :c_phone, :c_since, :c_credit, :c_credit_lim
            , :c_discount, :c_balance, :c_data :c_data_indicator, :h_date

        FROM TABLE ( PAY_C_LAST( :w_id
            , :d_id
            , :c_w_id
            , :c_d_id
            , :c_last_input
            , CAST(:h_amount AS DECIMAL(6,2))
            , :c_data_prefix_c_last
            )

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

            );

        COMMIT;

    END COMPOUND;

    }
    else
    {
        EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

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

        INTO :w_street_1, :w_street_2, :w_city, :w_state, :w_zip
            , :d_street_1, :d_street_2, :d_city, :d_state, :d_zip
            , :c_last, :c_first, :c_middle, :c_street_1, :c_street_2, :c_city, :c_state
            , :c_zip, :c_phone, :c_since, :c_credit, :c_credit_lim
            , :c_discount, :c_balance, :c_data :c_data_indicator, :h_date

```

```

FROM TABLE ( PAY_C_ID( :w_id
, :d_id
, :c_w_id
, :c_d_id
, :in_c_id
, CAST(h_amount AS DECIMAL(6,2))
, :c_data_prefix_c_id
)
) AS T( W_STREET_1, W_STREET_2, W_CITY, W_STATE, W_ZIP
, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
, C_LAST, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
, C_DISCOUNT, C_BALANCE, C_DATA, H_DATE
);

COMMIT ;

END COMPOUND ;

}

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

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

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

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

EXEC SQL ROLLBACK WORK ;

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

return ( clientRc );
}

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

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

EXEC SQL BEGIN DECLARE SECTION;

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

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

EXEC SQL END DECLARE SECTION;

```

```

int clientRc = TRAN_OK ;
int itemIndex = 0 ;

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

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

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

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

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

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

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

if ( sqlca.sqlcode == 0 )
{
// 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 ;
}

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

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

```

```

return ( clientRc );
}

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

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

EXEC SQL BEGIN DECLARE SECTION;

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

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

EXEC SQL END DECLARE SECTION;

int clientRc = TRAN_OK ;
int orderIndex = 0 ;

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

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

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

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

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

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

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

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

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

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

```

```

clientRc = FATAL_SQLERROR;
}

return ( clientRc );
}

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

#undef w_id
#undef d_id

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

int clientRc = TRAN_OK;

EXEC SQL BEGIN DECLARE SECTION;

// input

sqlint32  threshold;

// output

sqlint32  low_stock;

EXEC SQL END DECLARE SECTION;

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

stocklev->deadlocks = -1;
stocklev->s_transstatus = 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_ID ) INTO :low_stock

FROM ( SELECT DISTINCT S_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_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_transstatus = 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
#####

PRP_OPTS = PACKAGE \
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_DBG = tpcodbg$(OBJEXT)
UTIL_OBJ_GEN = tpcmisc$(OBJEXT)
UTIL_OBJ_DB2 = tpcctx$(OBJEXT)

```

```

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

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

dbggen: $(UTIL_OBJ_GEN)

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

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

connect:
- db2 connect to $(TPCC_DBNAME)

disconnect:
- db2 connect reset
- db2 terminate

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

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

.sqc.c:
@echo "Prepping $.sqc"
db2 prep $.sqc $(PRP_OPTS)
db2 grant execute on package TPCCCTX to public

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

# Source
tpcodbg$(OBJEXT): tpcodbg.c
tpcctx$(OBJEXT): tpcctx.c
tpcmisc$(OBJEXT): tpcmisc.c

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

```

### Src.Common/tpcctx.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.
**
*/
*
* tpcctx.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) {
            strcpy(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 */
    sqlintp(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, " |");
        for(k = 0; k < 16; k++) {
            int pos = j * 16 + k;
            char c = ' ';
            if (pos < 70) {
                c = psqlca->sqlerrmc[pos];
                if (!sprintf(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, "sqlerrd: ");
    for(j = 0; j < 6; j++)
        fprintf(err_fp, "%d", psqlca->sqlerrd[j]);
    fprintf(err_fp, "\n");

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

    fprintf(err_fp, "\n");

    fclose(err_fp);
}

/*-----*/
/* del_debug      */
/*-----*/

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

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

/*-----*/
/* del_print      */

```

```

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

    current_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");

    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, ")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_OL_CNT = %d (%X)\n",
in_neword->s_OL_CNT, in_neword->s_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_OL_CNT;
for (j=0; j<items; j++) {
if (j != 0)
fprintf(debug_fp, "\n");
fprintf(debug_fp, "tts_OL_ID[%d] = %d (%X)\n",
j, in_neword->in_item[j].s_OL_ID, in_neword->in_item[j].s_OL_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_OL_CNT = %d (%X)\n",
neword_ptr->s_OL_CNT, neword_ptr->s_OL_CNT);
fprintf(debug_fp, "ts_O_ENTRY_D = %s\n",
neword_ptr->s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_total_amount = %2f\n",
neword_ptr->s_total_amount);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
neword_ptr->deadlocks, neword_ptr->deadlocks);

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

fprintf(debug_fp, "titems {n");

```

```

items = neword_ptr->s_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] = %2f\n",
j, neword_ptr->item[j].s_I_PRICE);
fprintf(debug_fp, "tts_OL_AMOUNT[%d] = %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 i, 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, "t}\n\n");

fprintf(debug_fp, "out_ordstat_struct {n");
fprintf(debug_fp, "ts_C_ID = %d (%X)\n",

```

```

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

fprintf(debug_fp, "titems {n");
items = ordstat_ptr->s_ol_cnt;
for (j = 0; j < items; j++) {
if (j != 0)
fprintf(debug_fp, "\n");
fprintf(debug_fp, "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_ID[%d] = %d (%X)\n",
j, ordstat_ptr->item[j].s_OL_ID, ordstat_ptr->item[j].s_OL_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] = %2fn",
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, "t}\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,"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();
strcpy(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\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\n");
fclose(debug_fp);
}

void current_tmstamp(char *buf)
{
time_t t = time(NULL);

```

```

strcpy(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 - 2005
** 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 <windows.h>

#define RAND_A 16807
#define RAND_M 2147483647
#define RAND_M1 2147483646
#define RAND_MD 2147483647.0
#define RAND_Q 127773
#define RAND_R 2836

static int seed = 1;
static int seedflag = 0;

void random(int);
int random(void);
double current_time_ms(void);
double current_time(void);

void random (int initial_seed)
{
seed = initial_seed;
if ((seed < 1) || (seed > RAND_M1)) seed = 1;
}

int random (void)
{
int lo;
int hi;
int test;

hi = seed / RAND_Q;
lo = seed % RAND_Q;
test = RAND_A * lo - RAND_R * hi;
if (test > 0) seed = test;
else seed = test + RAND_M;

return (seed);
}

/* Current time in SECONDS, precision SECONDS */
double current_time(void)
{
/* truncate fractional seconds -> seconds */
return (double)((int)(current_time_ms()));
}

```

```

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
    /* GetCurrentTime() returns ms */
    /* convert to fractional seconds */
    return (GetCurrentTime() / 1000);
}

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 "ival.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 oiems_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;

```

```

};

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_transstatus;
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_transstatus;
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/drvtpcc.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.
*****/

/*
 * drvtpcc.h - Header file for the TPC-C batch driver
 */

#ifndef __DRVTPCC_H
#define __DRVTPCC_H

// Message formatting definitions
#define HELP_FMT " %-10st %-50s\n"
#define PRT_HDR "INUM Sys-Time ITime Bad Nord NordL NordR Nrj Del Drj Pay " \
"Prj OStat Orj SLev Srg Tpm-Cln" \
"-----" \
"-----\n"
#define PRT_FMT "%4d %8s %5d %3d %6d %6d %5d %3d %5d %3d %5d %3d %5d %3d %5d %3d %8.2fn"

#define BANNER_FMT "Comment:\n" \
"Warehouses:\n" \
"Warehouses Used:\n" \
"SP Type:\n" \
"NURand C Constants:\n" \
"Database Name:\n" \
"Rampup Time:\n" \
"Run Time:\n" \
"Interval Time:\n" \
"Transaction Type:\n" \
"Transaction Mix:\n" \
"Number of Transactions:\n" \
"Number of Clients:\n" \
"Run EMON:\n" \
"Output File:\n" \
"Raw Output File:\n" \
"\n"

#define MAXMIN_FMT "nMax./Min./Avg. TpmC => %2f (%d), %2f (%d), %2fn"
#define INTERVAL_FMT "Best %2d Min. Interval => %2f (%d-%d)\n"
#define MAXMIN_PER_WH_FMT "nMax./Min./Avg. TpmC per Warehouse => %2f, %2f, %2fn"

// Raw data message formatting definitions
#define RAW_HDR "INUM;Sys-Time;ITime;Bad;Nord;NordL;NordR;Nrj;Del;Drj;Pay;Prj; " \
"OStat;Orj;SLev;Srg;Tpm-Cln"
#define RAW_FMT "%4d;%8s;%5d;%3d;%6d;%6d;%5d;%3d;%5d;%3d;%5d;%3d;%5d;%3d;%5d;%3d;%5d;%3d;%8.2fn"
#define RAW_BANNER_FMT "Comment:\n" \
"Warehouses;\n" \
"Warehouses Used;\n" \
"SP Type;\n" \
"NURand C Constants:C_ID;%d;C_LAST;%d;OL_I_ID;%d\n" \
"Database Name;\n" \
"Rampup Time;%d;(seconds)\n" \
"Run Time;%d;(seconds)\n" \
"Interval Time;%d;(seconds)\n" \
"Transaction Type;%d;(s)\n" \
"Transaction Mix;New;%d;Pay;%d;Ord;%d;Del;%d;Stk;%d\n" \

```

```

"Number of Transactions;%d\n" \
"Number of Clients;%d\n" \
"Run EMON;%s\n" \
"Output File;%s\n" \
"Raw Output File;%s\n" \
"\n"

#define RAW_MAXMIN_FMT "nMax./Min./Avg. TpmC;%2f(%d);%2f(%d);%2fn"
#define RAW_INTERVAL_FMT "Best %2d Min. Interval;%2f(%d-%d)\n"
#define RAW_MAXMIN_PER_WH_FMT "nMax./Min./Avg. TpmC per Warehouse;%2f;%2f;%2fn"

// Interval statistics reporting format
#define ISTAT_FMT "%-12s %8d (%5.2f) %8d (%5.2f) %8d (%5.2f) %8.4f %8d (%5.2f)\n"
#define ISTAT_FMT2 "%-12s %8d (%5s) %8d (%5s) %8d (%5s) %8.4f %8d (%5s)\n"
#define ISTAT_LNSZ 100
#define NEWRATIO_FMT "NewOrders: Total/Remote/Ratio => %6d / %5d / %6.2f %3s\n" //AYL to report ratio
#define PAYRATIO_FMT "Payments: Total/Remote/Ratio => %6d / %5d / %6.2f %3s\n" //AYL to report ratio

/* Code from Porting.c */
/* Converted to Macros */
/* MTE 04/02/02 */

/* Code from Porting.h */
/* MTE 04/02/02 */

#ifndef FALSE
#define FALSE 0
#define TRUE 1
#endif

#define API_ERROR(x, y, z) fprintf(stderr, "%s(%d): %s failed - rc=%d\n", __FILE__, x, y, z); \
flush(stderr)

#ifndef WIN64
#define DWORD_PTR DWORD
#endif

#define nm_tStat_t 2 // Number of tStat structures required

typedef struct tInfo_t {
tStat_t stats[nm_tStat_t][STOCKLEV_SQL];
int count;
HANDLE tHandle;
DWORD tid;
} tInfo_t, *P_tInfo_t;

typedef struct SharedMem {
int sDone;
int Interval;
int Completed;

tInfo_t tInfo[1];
/*
 * do not define anything after tInfo in this structure
 * tInfo is allocated based upon the number of clients.
 * It appears that tInfo is an array of one, it is really dynamically
 * allocated memory which has numberOfClients positions allocated
 * in shared memory. Add variables to shared memory above tInfo.
 */
} SharedMem;

typedef struct fb_data {
double tpmCValue;
int startInterval;
int endInterval;
} fb_data;

#endif // __DRVTPCC_H

```

## include/lval.h

/\* lval.h - generated automatically at 20100315.1027 \*/

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 104040
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_ORDER 5
#define MAX_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 GetLastError()
```

```
/* *****
 * Windows I/O Macros */
*****
```

```
#ifndef INVALID_HANDLE_VALUE
#define INVALID_HANDLE_VALUE ((DWORD)-1)
#endif
```

```
#ifndef INVALID_SET_FILE_POINTER
#define INVALID_SET_FILE_POINTER ((DWORD)-1)
#endif
```

```
#define IOH_INIT(hnd, type, name) \
hnd->fd = INVALID_HANDLE_VALUE; \
hnd->type = type; \
hnd->name = name;
```

```
#define IOH_CREATE(hnd) \
if (hnd->type == IOH_PIPE) { \
    DWORD timeout = 1000; \
    hnd->fd = CreateNamedPipe(hnd->name, PIPE_ACCESS_OUTBOUND, \
        PIPE_TYPE_BYTE | PIPE_READMODE_BYTE | PIPE_WAIT, \
        1, 0, 0, timeout, NULL); \
    rc = (hnd->fd == INVALID_HANDLE_VALUE) ? -1 : 0; \
} else {
```

```
    rc = 0; \
} \
} \
#define IOH_OPEN(hnd) \
if (hnd->type == IOH_PIPE) { \
    rc = (ConnectNamedPipe(hnd->fd, NULL) != 0) ? 0 : -1; \
} else { \
    hnd->fd = CreateFile(hnd->name, GENERIC_WRITE, FILE_SHARE_WRITE, \
        NULL, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL); \
    rc = (hnd->fd == INVALID_HANDLE_VALUE) ? -1 : 0; \
    if (rc == 0 && hnd->type == IOH_FILE_APPEND) { \
        rc = SetFilePointer(hnd->fd, 0, 0, FILE_END); \
        if (rc == INVALID_SET_FILE_POINTER) { \
            rc = (GetLastError() == NO_ERROR) ? 0 : -1; \
        } else { \
            rc = 0; \
        } \
    } \
} \
}
```

```
#define IOH_WRITE(hnd, buff, num, num2) \
rc = (WriteFile(hnd->fd, buff, num, (LPDWORD)&num2, NULL) != 0) ? 0 : -1;
```

```
#define IOH_FLUSH(hnd) \
if (hnd->type == IOH_PIPE) { \
    rc = (FlushFileBuffers(hnd->fd) != 0) ? 0 : -1; \
} else { \
    rc = 0; \
} \
}
```

```
#define IOH_DELETE(hnd) rc = 0;
```

```
#define IOH_CLOSE(hnd) \
if (hnd->type == IOH_PIPE) { \
    rc = (DisconnectNamedPipe(hnd->fd) != 0) ? 0 : -1; \
    IOH_ERRMSG(hnd, "disconnecting"); \
    rc = (CloseHandle(hnd->fd) != 0) ? 0 : -1; \
} \
}
```

```
typedef DWORD IOH_NUM;
typedef HANDLE IOH_HND;
```

```
/* *****
 * Windows Semaphore Macros */
*****
```

```
#define SEM_HANDLE HANDLE
```

```
#define SEM_INIT(hnd, x, name) \
hnd = CreateSemaphore(NULL, x, 1, NULL); \
if (hnd == NULL) \
    API_ERROR(__LINE__, "CreateSemaphore", (rc=GEN_ERRCODE));
```

```
#define SEM_WAIT(hnd) \
if ((rc=WaitForSingleObject(hnd, INFINITE)) == WAIT_FAILED) \
    API_ERROR(__LINE__, "WaitForSingleObject", (rc=GEN_ERRCODE));
```

```
#define SEM_FREE(hnd) \
ReleaseSemaphore(hnd, 1, NULL)
```

```
#define SEM_DESTROY(hnd) \
if ((rc=CloseHandle(hnd)) == 0) \
    API_ERROR(__LINE__, "CloseHandle", (rc=GEN_ERRCODE));
```

```
/* *****
 * Common I/O Macros and Definitions */
*****
```

```
#define IOH_FILE 1
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3
```

```
#define IOH_ERRMSG(hnd, msg) \
if (rc != 0) {
```

```
    fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", GEN_ERRCODE, msg, \
        hnd->fd, hnd->type, hnd->name); \
    return rc; \
} \
}
```

```
struct _ioh {
    IOH_HND fd;
    int type;
    char *name;
};
```

```
typedef struct _ioh ioHandle;
```

```
/* *****
 * Generic I/O Routine Prototypes */
*****
```

```
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes);
int GenericClose(ioHandle *hnd);
```

```
/* *****
 * Generic I/O Routines */
*****
```

```
int GenericOpen(ioHandle *hnd, int type, char *name)
{
    int rc = 0;
```

```
    IOH_INIT(hnd, type, name)
```

```
    IOH_CREATE(hnd)
    IOH_ERRMSG(hnd, "creating")
```

```
    IOH_OPEN(hnd)
    IOH_ERRMSG(hnd, "opening")
```

```
    return rc;
```

```
}

int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes)
```

```
{
    int rc = 0;
    int numBytesWritten = -1;
```

```
    IOH_WRITE(hnd, Buffer, numBytes, numBytesWritten)
```

```
    IOH_ERRMSG(hnd, "writing")
```

```
    if (numBytes != numBytesWritten) {
        fprintf(stderr, "Truncated data writing to fd %d (%d, %s)\n", hnd->fd, hnd->type, hnd->name);
        rc = -1;
    }
    return rc;
```

```
}
```

```
int GenericClose(ioHandle *hnd)
```

```
{
    int rc = 0;
```

```
    IOH_FLUSH(hnd)
    IOH_ERRMSG(hnd, "flushing")
```

```
    IOH_CLOSE(hnd)
    IOH_ERRMSG(hnd, "closing")
```

```
    IOH_DELETE(hnd)
    IOH_ERRMSG(hnd, "deleting")
```

```
    return rc;
```

```
}

#endif // __PLATFORM_H
```

## include/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 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****
*/

/*
 * tpccapp.h - Application Macros
 */

#ifndef __TPCCAPP_H
#define __TPCCAPP_H

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

#include "sqlenv.h"
#define daricall __stdcall

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

#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var) SwapEndian((void*)&Var, sizeof(Var))
/*
*****
FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int i=0x12345678; SWAP_BYTE(i); i => 0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ 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; }

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

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

#endif // __TPCCAPP_H
```

## include/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
```

## 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 void random(int);
extern int random(void);

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

include/tpccutil.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.
*****/

```

```

/*
 * tpccutil.h - Miscellaneous prototypes and defines.
 */

#ifndef __TPCCUTIL_H
#define __TPCCUTIL_H

#ifdef _TPCCUTIL_C
#define EXTERN
#else
#define EXTERN extern
#endif

/*****
** Type definition section.
*****/
typedef void (*TPCC_FUNCTION)(void *, int);

typedef int (*TPCC_FUNC)(void *, void *);

typedef struct tStat_t {

    // ALL: Transaction Counters
    int trans, rejects, dlocks, _90, bad;
    double resptime;
    double keyTime, thinkTime;

    // NEWORDER: Local/Remote Counters
    int trans_l, trans_r;
    int rejects_l, rejects_r;
    int dlocks_l, dlocks_r;
    int _90_l, _90_r;
    double resptime_l, resptime_r;

    // NEWORDER: Good/Bad Counters
    int trans_g, trans_b;
    int rejects_g, rejects_b;
    int dlocks_g, dlocks_b;
    int _90_g, _90_b;
    double resptime_g, resptime_b;

    // DELIVERY: Skipped Orders/Items Counters
    int del_skip, ord_skip;

} tStat_t, *P_tStat_t;

#define sz_tStat_t sizeof(tStat_t)

// Stolen from Ivan Lew's code in db2stat.c
#define Li2Double(x) (((double)((x).HighPart) * 4.294967296E9 + (double)((x).LowPart)

/*****
** Externally available functions.
*****/

EXTERN void neword_sample(void *, int);
EXTERN void payment_sample(void *, int);
EXTERN void ordstat_sample(void *, int);
EXTERN void delivery_sample(void *, int);
EXTERN void stocklev_sample(void *, int);

EXTERN int run_measurement(int, P_tStat_t, int*, int*, int);

/*****
** Externally available variables.
*****/

#undef EXTERN

#endif // __TPCCUTIL_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=CK080131

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

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

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

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

set DB2VERSION=v9

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

@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_db2_last
set TPCC_DBNAME=TPCC
set TPCC_ROOT=c:\home\tpcc_db2_last\tpc-c\ibm
set TPCC_SQLLIB=c:\sqllib
set TPCC_RUNDATA=%HOME%\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 tpc.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
```

## tpccIsapi/resource.h

```
///  
// Microsoft Visual C++ generated include file.  
// Used by tpccIsapi.rc  
//  
#define IDS_PROJNAME 100
```

```
// Next default values for new objects  
//  
#ifndef APSTUDIO_INVOKED  
#ifndef APSTUDIO_READONLY_SYMBOLS  
#define _APS_NEXT_RESOURCE_VALUE 201  
#define _APS_NEXT_COMMAND_VALUE 32768  
#define _APS_NEXT_CONTROL_VALUE 201  
#define _APS_NEXT_SYMED_VALUE 101  
#endif  
#endif
```

## tpccIsapi/htmlPhraser.h

```
///  
// htmlPhraser.h  
// Class to decode a html query string  
//
```

```
#pragma once  
  
#include <memory.h>
```

```
///  
// Definitions  
//
```

```
#define NULL 0  
  
#define COMMAND_ID 0  
#define TERM_ID 1  
#define W_ID 2  
#define D_ID 3  
#define C_ID 4  
#define C_NAME 5  
  
#define C_W_ID 6  
#define C_D_ID 7  
#define AMT_PAID 8  
  
#define STK_THRESHOLD 9  
#define CARRIER_NUM 10  
  
#define ITEM_LIST_START 11  
#define ITEM_LIST_FINISH 55  
  
#define MAX_QUERY_ID 55  
#define MAX_FIELD_LEN 256  
#define MAX_FIELD_NUM 56
```

```
// Command Codes  
//  
#define NEW_ORDER_CODE 'n'  
#define PAYMENT_CODE 'p'  
#define ORDER_STATUS_CODE 'o'  
#define DELIVERY_CODE 'd'  
#define STOCK_CODE 's'  
#define EXIT_CODE 'e'  
#define MENU_CODE 'm'  
  
#define COMMAND_LOGIN 0  
#define COMMAND_NEW_ORDER 1  
#define COMMAND_PAYMENT 2  
#define COMMAND_ORDER_STATUS 3  
#define COMMAND_DELIVERY 4  
#define COMMAND_STOCK 5  
#define COMMAND_EXIT 6  
  
#define COMMAND_LOGIN_RESULTS 7  
#define COMMAND_NEW_ORDER_RESULTS 8  
#define COMMAND_PAYMENT_RESULTS 9  
#define COMMAND_ORDER_STATUS_RESULTS 10  
#define COMMAND_DELIVERY_RESULTS 11  
#define COMMAND_STOCK_RESULTS 12  
  
//  
// Class htmlPhraser  
//  
  
class htmlPhraser  
{  
public:  
    htmlPhraser(char *queryString);  
    ~htmlPhraser() {return;}  
  
    // getters  
public:  
    int getCommandId();  
    int validate(int bnType);  
  
    char * get_TERM_ID() {return iQueryValues[TERM_ID];}  
    char * get_W_ID() {return iQueryValues[W_ID];}  
    char * get_D_ID() {return iQueryValues[D_ID];}  
    char * get_C_ID() {return iQueryValues[C_ID];}  
    char * get_C_NAME() {return iQueryValues[C_NAME];}  
    char * get_C_W_ID() {return iQueryValues[C_W_ID];}  
    char * get_C_D_ID() {return iQueryValues[C_D_ID];}  
    char * get_AMT_PAID() {return iQueryValues[AMT_PAID];}  
    char * get_STK_THRESHOLD() {return  
iQueryValues[STK_THRESHOLD];}  
    char * get_CARRIER_NUM() {return iQueryValues[CARRIER_NUM];}  
  
    char * get_ITEM_SUPP_W(int item) {return iQueryValues[(ITEM_LIST_START + 0) +  
(item * 3)];}  
    char * get_ITEM_ITEM_NUM(int item) {return iQueryValues[(ITEM_LIST_START + 1) +  
(item * 3)];}  
    char * get_ITEM_QTY(int item) {return iQueryValues[(ITEM_LIST_START + 2) +  
(item * 3)];}  
  
    // Class Functions  
private:  
    char convertQueryToken(char **queryString);  
  
    // Class Attributes  
private:  
    int iCustomerIdFlag;  
    int iCarrierNumFlag;  
    int iStockThresholdFlag;  
  
    char iQueryValues[MAX_FIELD_NUM][MAX_FIELD_LEN];  
};
```

## tpccIsapi/StdAfx.cpp

```
// stdafx.cpp : source file that includes just the standard includes  
// tpccIsapi.pch will be the pre-compiled header  
// stdafx.obj will contain the pre-compiled type information
```

```
#include "stdafx.h"
```

```
// TODO: reference any additional headers you need in STDAFX.H  
// and not in this file
```

## tpccIsapi/htmlPhraser.cpp

```
///  
// htmlPhraser.cpp  
// Class implementation of htmlPhraser.  
// This class will take a query string and break it into a series  
// of constituent parts  
//
```

```
#include "htmlPhraser.h"
```

```
///  
// htmlPhraser::htmlPhraser  
// Title : Constructor  
// Parameters : char * query string  
// Return Value : None  
// Comments :  
//
```

```
htmlPhraser::htmlPhraser(char *queryString)  
{  
    // initialize query values  
    iCustomerIdFlag = iCarrierNumFlag = iStockThresholdFlag = false;  
  
    // this initializes the query list to NULL's. This means that  
    // characters being added are overwriting null characters and  
    // therefore the string will be null terminated implicitly.  
  
    memset(iQueryValues,NULL,(MAX_FIELD_NUM * MAX_FIELD_LEN));  
  
    // controls  
    char queryChar = NULL;  
  
    int queryIndex = -1;  
    int valueIndex = -1;  
  
    // process each character of query string  
    while(*queryString)  
    {  
        // check for special case characters  
        if(queryChar)  
        {  
            // a percentage sign would indicate a token  
            if(*queryString != '%')  
            {  
                // a plus sign represents a space  
                if(*queryString == '+')  
                {  
                    queryChar = ' ';  
                    *queryString++;  
                }  
                else queryChar = *queryString++;  
            }  
        }  
    }  
}
```

```

        else queryChar = convertQueryToken(&queryString);
    }
    else queryChar = '&';

    // handle query reference (&)
    if(queryChar == '&')
    {
        // reset value index
        valueIndex = -1;

        // do we have a numeric query reference
        if(*queryString >= '0' && *queryString <= '9')
        {
            // numeric query id
            queryIndex =
                ((*queryString - '0') * 10) + ((*queryString + 1) - '0');

            // walk past the two command characters
            queryString += 2;

            // validate query value
            if(queryIndex > MAX_QUERY_ID)
                queryIndex = -1;
        }
        else queryIndex = -1;

        // finished processing for query reference
        continue;
    }

    // we have a query reference but need to wait until we see '='
    // before accepting value
    if(valueIndex == -1)
    {
        // we are waiting for '='
        if(queryChar == '=')
        {
            valueIndex = 0;

            // set query string flags
            switch(queryIndex)
            {
                case C_ID:
                    iCustomerIdFlag = true; break;
                case CARRIER_NUM:
                    iCarrierNumFlag = true; break;
                case STK_THRESHOLD:
                    iStockThresholdFlag = true; break;
                default: break;
            }
        }

        // finishes looging for '='
        continue;
    }

    // add each character to the query value
    if(queryIndex > -1 && valueIndex > -1)
    {
        // we are processing a query value
        if(valueIndex < MAX_FIELD_LEN)
        {
            // we have not exceeded max line len
            iQueryValues[queryIndex][valueIndex++] = queryChar;
        }
        continue;
    }
}

return;
}

```

```

////////////////////////////////////
// htmlPhraser::getCommandId
////////////////////////////////////
// Title   : Returns the page command
// Parameters   : None
// Return Value : int - page command
// Comments   :
////////////////////////////////////

int htmlPhraser::getCommandId()
{
    // return command numeric code
    switch(*iQueryValues[COMMAND_ID])
    {
        case NEW_ORDER_CODE:
            if(iCustomerIdFlag)
                return COMMAND_NEW_ORDER_RESULTS;
            else return COMMAND_NEW_ORDER;
        case PAYMENT_CODE:
            if(iCustomerIdFlag)
                return COMMAND_PAYMENT_RESULTS;
            else return COMMAND_PAYMENT;
        case ORDER_STATUS_CODE:
            if(iCustomerIdFlag)
                return COMMAND_ORDER_STATUS_RESULTS;
            else return COMMAND_ORDER_STATUS;
        case DELIVERY_CODE:
            if(iCarrierNumFlag)
                return COMMAND_DELIVERY_RESULTS;
            else return COMMAND_DELIVERY;
        case STOCK_CODE:
            if(iStockThresholdFlag)
                return COMMAND_STOCK_RESULTS;
            else return COMMAND_STOCK;
        case MENU_CODE:
            return COMMAND_LOGIN_RESULTS;
        case EXIT_CODE:
            return COMMAND_EXIT;
        default:
            return COMMAND_LOGIN;
    };

    // should not get here
    return COMMAND_LOGIN;
}

////////////////////////////////////
// htmlPhraser::validate
////////////////////////////////////
// Title   : validate url parameter list for all txn types
// Parameters   : int - txn type
// Return Value : int - error code
// Comments   :
////////////////////////////////////

int validate(int txnType)
{
    return 0;
}

////////////////////////////////////
// htmlPhraser::convertQueryToken
////////////////////////////////////
// Title   : Returns the page command
// Parameters   : None
// Return Value : int - page command
// Comments   :
////////////////////////////////////

char htmlPhraser::convertQueryToken(char **queryString)
{

```

```

char queryChar = NULL;

// skip over %
(*queryString)++;

// look at first character
switch(**queryString)
{
    case '2':
        {
            // what follows?
            (*queryString)++;

            switch(**queryString)
            {
                case '1':
                    queryChar = '!';
                    break;
                case '3':
                    queryChar = '#';
                    break;
                case '4':
                    queryChar = '$';
                    break;
                case '5':
                    queryChar = '%';
                    break;
                case '6':
                    queryChar = '&';
                    break;
                case '8':
                    queryChar = '(';
                    break;
                case '9':
                    queryChar = ')';
                    break;
                case 'B':
                    queryChar = '+';
                    break;
                case 'C':
                    queryChar = ',';
                    break;
                case 'F':
                    queryChar = '/';
                    break;
                case ' ':
                    queryChar = ' ';
                    break;
            }
        }
        break;
    case '3':
        {
            // what follows?
            (*queryString)++;

            switch(**queryString)
            {
                case 'A':
                    queryChar = '@';
                    break;
                case 'B':
                    queryChar = ' ';
                    break;
                case 'D':
                    queryChar = '=';
                    break;
                case 'F':
                    queryChar = '?';
                    break;
                case ' ':
                    queryChar = ' ';
            }
        }
    }
}

```

```

        break;
    }
}
break;
case '4':
{
    // what follows?
    (*queryString)++;

    switch(**queryString)
    {
        case '0':
            queryChar = '@';
            break;
        case '1':
            queryChar = ' ';
            break;
    }
}
break;
case '5':
{
    // what follows?
    (*queryString)++;

    switch(**queryString)
    {
        case 'B':
            queryChar = '[';
            break;
        case 'D':
            queryChar = ']';
            break;
        case 'E':
            queryChar = '^';
            break;
        case ' ':
            queryChar = ' ';
            break;
    }
}
break;
case '7':
{
    // what follows?
    (*queryString)++;

    switch(**queryString)
    {
        case 'B':
            queryChar = '[';
            break;
        case 'C':
            queryChar = '|';
            break;
        case 'D':
            queryChar = '>';
            break;
        case 'E':
            queryChar = '~';
            break;
        case ' ':
            queryChar = ' ';
            break;
    }
}
break;
case '+':
    queryChar = '+';

```

```

        break;
    }
}
// advance pointer and return
(*queryString)++; return queryChar;
}

```

```

////////////////////////////////////

```

### tpccIsapi/time.cpp

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/timeb.h>
#include <time.h>

char *get_time_prefix(char *buffer)
{
    time_t cur_time;
    char time_str[30];
    int len;

    cur_time = time(&cur_time);
    strftime(time_str, 29, "%X", localtime(&cur_time));

    len = sprintf(buffer, "%s - ",
                 time_str);
    if (len >= 30) {
        sprintf(buffer, "too small: %d\n",
                30, len);
    }
    return(buffer);
}

```

### tpccIsapi/StdAfx.h

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#pragma once

#define WIN32_LEAN_AND_MEAN // Exclude rarely-used stuff from Windows headers

#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some CString constructors will be explicit

// turns off ATL's hiding of some common and often safely ignored warning messages
#define _ATL_ALL_WARNINGS

// critical error descriptions will only be shown to the user
// in debug builds. they will always be logged to the event log
#ifdef _DEBUG
#define ATL_CRITICAL_ISAPI_ERROR_LOGONLY
#endif

#ifdef _WIN32_WINNT
#define _WIN32_WINNT 0x0403
#endif

// TODO: this disables support for registering COM objects
// exported by this project since the project contains no
// COM objects or typelib. If you wish to export COM objects
// from this project, add a typelib and remove this line
#define _ATL_NO_COM_SUPPORT

#include "resource.h"

```

```

#include <atlsrvres.h>
#include <atlisapi.h>
#include <atlstencil.h>

// TODO: reference additional headers your program requires here

```

### tpccIsapi/tpcc.h

```

// Common defines and structures use internally by client code
// Not to be confused with structures actually passed in transactions
//

```

```

// standard includes

```

```

#ifndef _COMMON_TPCC
#define _COMMON_TPCC

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/timeb.h>
#include <time.h>

```

```

#ifdef DB2
#include <db2tpcc.h>
#endif

```

```

#ifdef ORACLE
#include <ora_tpcc.h>
#endif

```

```

#ifdef SYBASE
#include <sybtpcc.h>
#endif

```

```

#include <iostream>
#include <fstream>
#include <process.h>
#include <ios>

```

```

////////////////////////////////////
// Defines
////////////////////////////////////

```

```

#define OK 0
#define INVALID_STATUS -1
#define ERR -1
#define INVALID_COM_STATUS -2

```

```

#define TXN_MAX_COMMANDS 55
#define MAX_TRANSACTIONS 14
#define MAX_CMD_LENGTH 100
#define INPUT_ITEMS 3
#define MAX_INT_BUFFER 15
#define NORD_ITEMS 15
#define ITEM_START 11
#define ITEM_END 55
#define MAX_ITEMS 15

```

```

#define MAX_STRING_LEN 256
#define MAX_HTML_PAGE_LEN 4096
#define MAX_HTML_HEADER_LEN 512

```

```

#define DELIVERY_THREADS_NUM 100

```

```

#define DISTRICTS_PER_WAREHOUSE 10

```

```

////////////////////////////////////
// Transaction Codes
////////////////////////////////////

```

```

#define TXN_LOGIN 0
#define TXN_NEW_ORDER 1 2
#define TXN_PAYMENT 2
#define TXN_ORDER_STATUS 3 4
#define TXN_DELIVERY 4
#define TXN_STOCK 5
#define TXN_EXIT 6
#define TXN_LOGIN_RESULTS 7
#define TXN_NEW_ORDER_RESULTS 8
#define TXN_PAYMENT_RESULTS 9
#define TXN_ORDER_STATUS_RESULTS 10
#define TXN_DELIVERY_RESULTS 11
#define TXN_STOCK_RESULTS 12

#define CMD_NORD "nord"
#define CMD_PYMT "pymt"
#define CMD_ORDS "ords"
#define CMD_DLVY "dlvy"
#define CMD_STOK "stok"
#define CMD_EXIT "exit"
#define CMD_MENU "menu"

#define APP_NAME "tpcc.html"
#define HEADER "Content-
Type:text/html\r\nContent-Length: %d\r\nConnection: Keep-Alive\r\n\r\n"

// URL Commands
// URL Commands

#define CMD_TXN_ID "00"
#define CMD_TERM_ID "01"
#define CMD_W_ID "02"
#define CMD_D_ID "03"
#define CMD_C_ID "04"
#define CMD_C_NAME "05"
#define CMD_C_W_ID "06"
#define CMD_C_D_ID "07"
#define CMD_AMT_PAID "08"
#define CMD_STK_THRESHOLD "09"
#define CMD_CARRIER_NUM "10"

#define ITEM01_SUPP_W "11"
#define ITEM01_ITEM_NUM "12"
#define ITEM01_OTY "13"

#define CHAR_FILL ''
#define NUMERIC_FILL ''
#define NEGITIVE_SYMBOL '-'
#define MONEY_SYMBOL '$'
#define DECIMAL_SYMBOL '.'
#define ZERO_SYMBOL '0'
#define ZIP_DELIMITER ':'
#define PHONE_DELIMITER '-'
#define DATE_DELIMITER '-'
#define TIME_DELIMITER ':'

#define DEFAULT_MONEY64_LEN 15
#define DEFAULT_MONEY32_LEN 9
#define DEFAULT_MONEY16_LEN 9

#define DEFAULT_NUMERIC64_LEN 15
#define DEFAULT_NUMERIC32_LEN 9
#define DEFAULT_NUMERIC16_LEN 9

#define DEFAULT_DECIMAL64_LEN 5
#define DEFAULT_DECIMAL32_LEN 5
#define DEFAULT_DECIMAL16_LEN 5

#define DEFAULT_DATETIME_LEN 19
#define DEFAULT_DATE_LEN 11

#define DEFAULT_TIME_LEN 8
#define DEFAULT_STRING_LEN 25
#define DEFAULT_ZIP_LEN 17
#define DEFAULT_PHONE_LEN 18

// String Field Lengths
// String Field Lengths

#define NAME_LEN 24
#define LAST_NAME_LEN 16
#define FIRST_NAME_LEN 16
#define INITIALS_LEN 2

#define CREDIT_LEN 2

#define STREET_LEN 20
#define CITY_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9

#define PHONE_LEN 16
#define DATA_LEN 200

#define ITEM_LIST 15
#define ORDER_LIST 10

// Type definitions
// Type definitions

typedef __int8 INT8b;
typedef __int16 INT16b;
typedef __int32 INT32b;
typedef __int64 INT64b;

typedef unsigned __int8 UINT8b;
typedef unsigned __int16 UINT16b;
typedef unsigned __int32 UINT32b;
typedef unsigned __int64 UINT64b;

typedef INT16b sqint16;
typedef INT32b sqint32;
typedef INT64b sqint64;

typedef INT16b int16_t;
typedef INT32b int32_t;
typedef INT64b int64_t;

typedef char BYTE8b;
typedef double DOUBLE;
typedef unsigned long NATURAL;

// Date and time values
// Date and time values

#define SECONDS_IN_DAY 86400
#define SECONDS_IN_HOUR 3600
#define SECONDS_IN_MINUTE 60
#define GMT_OFFSET 5

#define DAYS_IN_YEAR 365
#define YEARS_IN_LEAP 4
#define START_YEAR 1970
#define MONTHS_IN_YEAR 12

// Error codes
// Error codes

#define ERR_INVALID_TXN_TYPE -1

#define ERR_MISSING_W_ID -2
#define ERR_NON_NUMERIC_W_ID -3
#define ERR_MISSING_D_ID -4
#define ERR_NON_NUMERIC_D_ID -5
#define ERR_MISSING_C_ID -6
#define ERR_NON_NUMERIC_C_ID -7

#define ERR_MISSING_SUPP_W -8
#define ERR_NON_NUMERIC_SUPP_W -9
#define ERR_MISSING_ITEM_NUM -10
#define ERR_NON_NUMERIC_ITEM_NUM -11
#define ERR_MISSING_ITEM_OTY -12
#define ERR_NON_NUMERIC_ITEM_QTY -13

#define ERR_MISSING_CLAST_NAME -14
#define ERR_NON_NUMERIC_CUST_W_ID -15
#define ERR_NON_NUMERIC_CUST_D_ID -16
#define ERR_MISSING_AMOUNT_PAID -17
#define ERR_NON_NUMERIC_AMOUNT_PAID -18

#define ERR_INVALID_D_ID "ERROR : Invalid District ID. Try Again."
#define ERR_INVALID_W_ID "ERROR : Invalid Warehouse ID. Try Again."
#define ERR_INVALID_C_ID "ERROR : Invalid Customer ID. Try Again."
#define ERR_INVALID_SUPPLY_W_ID "ERROR : Invalid Item Supply Warehouse. Try Again."
#define ERR_INVALID_ITEM_NUM "ERROR : Invalid Item Number. Try Again."
#define ERR_INVALID_ITEM_OTY "ERROR : Invalid Item Qty. Try Again."
#define ERR_MISSING_C_ID_OR_CLAST "ERROR : Must Enter Customer Id or Customer Last Name. Try Again."
#define ERR_INVALID_PAYMENT_AMOUNT "ERROR : Invalid Payment Amount. Try Again."
#define ERR_INVALID_CARRIER "ERROR : Invalid Carrier Number. Try Again."
#define ERR_INVALID_THRESHOLD "ERROR : Invalid Threshold. Try Again."
#define ERR_INVALID_C_D_ID "ERROR : Invalid Customer District Id. Try Again."
#define ERR_INVALID_C_W_ID "ERROR : Invalid Customer Warehouse Id. Try Again."
#define ERR_TERMINAL_FULL "ERROR : Terminal can not support user. Terminal full."
#define ERR_C_ID_OR_CLAST_ONLY "ERROR : Either customer id or customer last name can be specified."

#define ERR_UNABLE_TO_OPEN_REG -50
#define ERR_DLVY_THREAD_FAILED -51
#define ERR_DLVY_SEMAPHORE_INIT_FAILED -52
#define ERR_DLVY_EVENT_INIT_FAILED -53
#define ERR_DLVY_QUEUE_EATING_TAIL -54
#define ERR_DLVY_QUEUE_CALLOC_FAIL -55

#define ERR_INVALID_USERNAME -70
#define ERR_INVALID_PASSWORD -71
#define ERR_INVALID_DB_NAME -72
#define ERR_INVALID_REGISTRY_KEY -73
#define ERR_DB2_DLL_NOT_LOADED -74
#define ERR_ORACLE_DLL_NOT_LOADED -75
#define ERR_CONNECT_ADDRESS_NOT_FOUND -76
#define ERR_NORD_ADDRESS_NOT_FOUND -77
#define ERR_PYMT_ADDRESS_NOT_FOUND -78
#define ERR_ORDS_ADDRESS_NOT_FOUND -79
#define ERR_DLVY_ADDRESS_NOT_FOUND -80
#define ERR_STOK_ADDRESS_NOT_FOUND -81
#define ERR_NULL_DLL_NOT_LOADED -82
#define ERR_UNKNOWN_DB -83
#define ERR_DISCONNECT_ADDRESS_NOT_FOUND -84
#define ERR_ORA_DLL_NOT_LOADED -85
#define ERR_SYB_DLL_NOT_LOADED -86
#define ERR_DBINIT_ADDRESS_NOT_FOUND -87

#define ERR_SAVING_CONTEXT -90
#define ERR_DETACHING_CONTEXT -91
#define ERR_ATTACHING_CONTEXT -92
#define ERR_HANDLE_IN_USE -93

#define ERR_CONNECT_TO_TM_FAILED -99
#define ERR_DLVY_LOG_OPEN_FAILED -100
#define ERR_DLVY_QUEUE_FULL -101

```



```

// Registry Definitions
////////////////////////////////////////////////////////////////////
#define REGISTRY_SUB_KEY"SOFTWARE\TPCC"

#define DELIVERY_THREADS "divyThreads"
#define DELIVERY_QUEUE_LEN "divyQueueLen"
#define DELIVERY_LOG_PATH "divyLogPath"
#define ERROR_LOG_FILE "errorLogFile"
#define HTML_TRACE_LOG_FILE "htmlTraceLogFile"
#define DB_NAME "dbName"
#define NULL_DB "nullDB"
#define COM_NULL_DB "commnullDB"
#define CLIENT_NULL_DB "clientNullDB"

#define NUM_USERS "numUsers"
#define DB_TYPE "dbType"

#define TXN_MONITOR "txn_server"
#define COMM_POOL "comm_pool"
#define HTML_TRACE "htmlTrace"
#define ISAPI_TRACE "isapi_trace"

#define DEFAULT_DLVY_THREADS 1
#define DEFAULT_DLVY_QUEUE_LEN 10
#define DEFAULT_DLVY_LOG_PATH "c:\\inetpub\\wwwroot\\tpcc\\divy"
#define DEFAULT_ERROR_LOG_FILE "c:\\inetpub\\wwwroot\\tpcc\\errorLog.txt"
#define DEFAULT_HTML_TRACE_LOG_FILE "c:\\inetpub\\wwwroot\\tpcc\\htmlTrace.txt"
#define DEFAULT_NUM_USERS 10000

#define DEFAULT_DB_NAME "tpcc"

////////////////////////////////////////////////////////////////////
// Structure defines
////////////////////////////////////////////////////////////////////

struct nord_wrapper {
    in_neword_struct in_nord;
    out_neword_struct out_nord;
};

struct paym_wrapper {
    struct in_payment_struct in_paym;
    struct out_payment_struct out_paym;
};

struct ords_wrapper {
    struct in_ordstat_struct in_ords;
    struct out_ordstat_struct out_ords;
};

struct divy_wrapper {
    struct in_delivery_struct in_divy;
    struct out_delivery_struct out_divy;
};

struct stok_wrapper {
    struct in_stocklev_struct in_stok;
    struct out_stocklev_struct out_stok;
};

typedef struct
{
    int year;
    int month;
    int day;

    int hour;
    int minute;
    int second;
} datetime;

struct NEWORDERDATA
{

```

```

    struct in_items_struct {
        int s_OL_I_ID;
        int s_OL_SUPPLY_W_ID;
        short s_OL_QUANTITY;
    } in_item[15];

    long long in_s_O_ENTRY_D_time; /* init by SUT */
    int in_s_C_ID;
    int in_s_W_ID;
    short in_s_D_ID;
    short in_s_O_OL_CNT; /* init by SUT */
    short in_s_all_local;
    short in_duplicate_items;

    struct out_items_struct {
        float s_I_PRICE;
        float s_OL_AMOUNT;
        short s_S_QUANTITY;
        char s_I_NAME[25];
        char s_brand_generic;
    } out_item[15];

    long long out_s_O_ENTRY_D_time;
    double out_s_W_TAX;
    double out_s_D_TAX;
    double out_s_C_DISCOUNT;
    double out_s_total_amount;
    int out_s_O_ID;
    short out_s_O_OL_CNT;
    short out_s_transtatus;
    short out_deadlocks;
    char out_s_C_LAST[17];
    char out_s_C_CREDIT[3];
};

struct PAYMENTDATA
{
    long long in_s_H_DATE_time;
    double in_s_H_AMOUNT;
    int in_s_W_ID;
    int in_s_C_W_ID;
    int in_s_C_ID;
    short in_s_C_D_ID;
    short in_s_D_ID;
    char in_s_C_LAST[17];

    long long out_s_H_DATE_time;
    long long out_s_C_SINCE_time;
    double out_s_C_CREDIT_LIM;
    double out_s_C_BALANCE;
    double out_s_C_DISCOUNT;
    int out_s_C_ID;
    short out_s_transtatus;
    short out_deadlocks;
    char out_s_W_STREET_1[21];
    char out_s_W_STREET_2[21];
    char out_s_W_CITY[21];
    char out_s_W_STATE[3];
    char out_s_W_ZIP[10];
    char out_s_D_STREET_1[21];
    char out_s_D_STREET_2[21];
    char out_s_D_CITY[21];
    char out_s_D_STATE[3];
    char out_s_D_ZIP[10];
    char out_s_C_FIRST[17];
    char out_s_C_MIDDLE[3];
    char out_s_C_LAST[17];
    char out_s_C_STREET_1[21];
    char out_s_C_STREET_2[21];
    char out_s_C_CITY[21];
    char out_s_C_STATE[3];
    char out_s_C_ZIP[10];
};

```

```

    char out_s_C_PHONE[17];
    char out_s_C_CREDIT[3];
    char out_s_C_DATA[201];
};

struct ORDERSTATUSDATA
{
    int in_s_C_ID;
    int in_s_W_ID;
    short in_s_D_ID;
    char in_s_C_LAST[17];

    double out_s_C_BALANCE;
    long long out_s_O_ENTRY_D_time;
    int out_s_C_ID;
    int out_s_O_ID;
    short out_s_O_CARRIER_ID;
    short out_s_ol_cnt;
    struct out_oitems_struct {
        long long s_OL_DELIVERY_D_time;
        float s_OL_AMOUNT;
        int s_OL_I_ID;
        int s_OL_SUPPLY_W_ID;
        short s_OL_QUANTITY;
    } out_item[15];
    short out_s_transtatus;
    short out_deadlocks;
    char out_s_C_FIRST[17];
    char out_s_C_MIDDLE[3];
    char out_s_C_LAST[17];
};

struct DELIVERYDATA
{
    long long in_s_O_DELIVERY_D_time;
    int in_s_W_ID;
    short in_s_O_CARRIER_ID;
    int out_s_O_ID[10];
    short out_s_transtatus;
    short out_deadlocks;
};

struct STOCKLEVELDATA
{
    int in_s_threshold;
    int in_s_W_ID;
    short in_s_D_ID;

    int out_s_low_stock;
    short out_s_transtatus;
    short out_deadlocks;
};

struct DLVYQUEUEDATA
{
    int warehouse;
    short in_s_O_CARRIER_ID;

    struct _timeb enqueueTime;
};

// MISCELLANEOUS HELPER FUNCTIONS
inline void appendText(char *string, char *text);
inline void appendText(char **string, char *text, int length, int justify);
inline void appendChar(char **string, char byte);
inline void DEBUGMSG(FILE * debugFile, char * message);
inline void appendSpaces(char **string, int spaces);

inline void calcOutDateTime(const INT64b value, datetime *timestamp);
inline int copyOutPhone(char *buffer, char *value, int len);
inline bool copyInMoney64(const char * value, INT64b *number);

```

```

inline bool copyInMoney32(const char * value,int *number);
inline int copyInMoney(const char *value);
inline void copyOutMoney64(char *buffer,INT64b value,unsigned int len);
inline int copyOutDate(char *buffer,INT64b value);
inline int copyOutDate(char *buffer,INT64b value);
inline int copyOutTime(char *buffer,INT64b value);
inline int copyOutDecimal64(char *buffer,INT64b value,unsigned int len);

```

```

inline UINT16b changeOrder16(UINT16b value);
inline UINT32b changeOrder32(UINT32b value);
inline UINT64b changeOrder64(UINT64b value);

```

```

inline INT16b changeOrder16(INT16b value);
inline INT32b changeOrder32(INT32b value);
inline INT64b changeOrder64(INT64b value);

```

```

//
// Name      : appendText
// Description :
// Append text to string
// Parameters :
// char ** - string point to append to
// char * - text to append
// Returns   :
// None
// Comments  :
//

```

```

inline void appendText(char **string,char *text)
{
    while(*text)
    {
        *(*string)++ = *text++;
    }

    **string='\0';
    return;
}

```

```

//
// Name      : appendText
// Description :
// Append text to string
// Parameters :
// char ** - string point to append to
// char * - text to append
// int - total field length including blank spaces
// int - justify flag
// Returns   :
// None
// Comments  :
// right justify
// left justify
//

```

```

inline void appendText(char **string,char *text,int length,int justify)
{
    int byteCount = 0;

    if(justify)
    {
        while(*text)
        {
            *(*string)++ = *text++;
            if(++byteCount > length)
                break;
        }

        //append blank spaces if text is less than length at end
        for(byteCount;byteCount < length;byteCount++)
            *(*string)++ = ' ';
    }
    else

```

```

{
    long long textLen = strlen(text);
    for(textLen;textLen < length;textLen++)
        *(*string)++ = ' ';

    while(*text)
        *(*string)++ = *text++;
}
**string='\0';
}

```

```

// Name      : appendChar
// Description :
// Append text to string
// Parameters :
// char ** - string point to append to
// char * - text to append
// Returns   :
// None
// Comments  :
//

```

```

inline void appendChar(char **string,char byte)
{
    *(*string)++ = byte;
    **string='\0';

    return;
}

```

```

//
// Name      : appendSpaces
// Description :
// appends buffer spaces to result page
// Parameters :
// **htmlPage
// Returns   :
// amount of characters the function appened
// to the html page
// Comments  :
//

```

```

inline void appendSpaces(char **string,int spaces)
{
    for(int index=0;index<spaces;index++)
    {
        *(*string)++ = ' ';
    }

    **string='\0';
}

```

```

//
// Name      : appendCustData
// Description :
// appends cust data buffer to result page
// Parameters :
// **htmlPage
// Returns   :
// Adds a newline character every 50 characters displayed.
// Comments  :
//

```

```

inline void appendCustData(char **string,char *text)
{
    short byteCount = 0;
    while(*text)

```

```

{
    *(*string)++ = *text++;
    byteCount++;
    if((byteCount % 50) == 0)
    {
        *(*string)++ = '\n';
        *(*string)++ = ' '; *(*string)++ = ' '; *(*string)++ = ' '; *(*string)++ = ' ';
        *(*string)++ = ' '; *(*string)++ = ' '; *(*string)++ = ' '; *(*string)++ = ' ';
        *(*string)++ = ' '; *(*string)++ = ' '; *(*string)++ = ' ';
    }
}
**string='\0';
}
}

```

```

//
// calcOutDateTime
//
// Title      : Calculate date & time data out of class array
// Parameters : INT64b - date & time expressed in seconds
// datetime * - timestamp
// Return Value : None
// Comments  :
//

```

```

inline void calcOutDateTime(const INT64b value,datetime *timestamp)
{
    // fixed days in each month (FEB 29 is special case)
    static int daysInMonth[12] = {31,28,31,30,31,30,31,31,30,31,30,31};

    // mask out EPOCH seconds
    int dateValue = ((int) (value & 0xffffffff)) +
        (SECONDS_IN_DAY - (GMT_OFFSET * SECONDS_IN_HOUR));

    int offset = (int) (value >> 32);

    // break out the seconds
    int hms = dateValue % SECONDS_IN_DAY;
    int days = dateValue / SECONDS_IN_DAY;

    int years = (days - 1) / DAYS_IN_YEAR;
    int leaps = years / YEARS_IN_LEAP;

    int daysUsed = (years * DAYS_IN_YEAR) + leaps;

    // adjust the number of days to account for calculated years
    days = days - daysUsed;

    // set the starting year, month, and day
    timestamp->day = 1;
    timestamp->month = 1;
    timestamp->year = START_YEAR + years;

    // is the current year a leap year
    int leap = !(timestamp->year % YEARS_IN_LEAP);

    // apply remaining days based on days in months
    int daysInCurrentMonth;

    while(days)
    {
        // get days in current month
        daysInCurrentMonth = daysInMonth[(timestamp->month - 1)];
        if(timestamp->month == 2 && leap)
            daysInCurrentMonth = daysInCurrentMonth + 1;

        // days > days in current month
        if(days > daysInCurrentMonth)
        {
            // increment month
            timestamp->month += 1;
            days = days - daysInCurrentMonth;

```

```

// month exceeds months in year
if(timestamp->month > MONTHS_IN_YEAR)
{
    // increment year and reset month
    timestamp->year += 1; timestamp->month = 1;

    // are we now on a leap year
    leap = !(timestamp->year % YEARS_IN_LEAP);
}
else
{
    // set day of month to remaining days
    timestamp->day = days; days = 0;
}

// set time values to remaining seconds
timestamp->hour = hms / SECONDS_IN_HOUR;
hms = hms % SECONDS_IN_HOUR;

timestamp->minute = hms / SECONDS_IN_MINUTE;
timestamp->second = hms % SECONDS_IN_MINUTE;
return;
}

//
// copyOutZip
//
// Title      : Copy zip data out of class array
// Parameters : char * - buffer to copy zip string into
//
// Return Value : int - Length of copy
// Comments    :
//
inline int copyOutZip(char *buffer, char *value, int len = DEFAULT_ZIP_LEN)
{
    int index      = 0;
    int bufferPos  = 0;

    // add each digit of zip number to buffer inserting delimiter at 5
    while(value[index] && bufferPos < len)
    {
        if(index == 5)
            buffer[bufferPos++] = ZIP_DELIMITER;

        buffer[bufferPos++] = value[index++];
    }

    // space fill to the required length
    while(bufferPos < len)
        buffer[bufferPos++] = CHAR_FILL;

    buffer[bufferPos] = NULL;
    return len;
}

//
// copyOutPhone
//
// Title      : Copy phone data out of class array
// Parameters : char * - buffer to copy phone string into
//
// Return Value : int - Length of copy
// Comments    :
//
inline int copyOutPhone(char *buffer, char *value, int len = DEFAULT_PHONE_LEN)
{
    int index      = 0;
    int bufferPos  = 0;

```

```

// add each digit of phone number to buffer inserting delimiter before 6, 9, and 12
while(value[index] && index < len)
{
    switch(index)
    {
        case 6:
        case 9:
        case 12:
            // insert delimiter
            buffer[bufferPos++] = PHONE_DELIMITER;
        default:
            // add phone digit to buffer
            buffer[bufferPos++] = value[index++];
    }
}

// space fill to the required length
while(bufferPos < len)
    buffer[bufferPos++] = CHAR_FILL;

buffer[bufferPos] = '\0';
return len;
}

//
// copyInMoney64
//
// Title      : Copy money data into class array
// Parameters : const char * - value string
// Return Value : INT64b integer value
// Comments    :
//
inline bool copyInMoney64(const char * value, INT64b *number)
{
    //INT64b  number      = 0;
    int      index      = 0;
    int      decimal     = 0;
    int      decimals    = 0;
    int      digitsAfterDec = 0;

    bool     negativeFlag = false;

    // convert each digit to a numeric portion
    while(value[index])
    {
        // handle $ . - All the rest assumed numeric
        switch(value[index])
        {
            case MONEY_SYMBOL:
                // ignore $ sign
                break;
            case NEGITIVE_SYMBOL:
                // set negative flag
                negativeFlag = true;
                break;
            case DECIMAL_SYMBOL:
                // set decimal
                decimal=1;
                decimals++;
                if(decimals > 1)
                    //more than 1 decimal point found
                    return false;
                break;
            default:
                // adjust decimal places
                decimal = decimal * 10;

                // add digit to running total

```

```

if(value[index] >= '0' && value[index] <= '9')
{
    if(decimal)
        if(++digitsAfterDec > 2)
            return false;

    *number = (*number * 10) + (value[index] - '0');
}
else
{
    //non-numeric field inserted
    return false;
}
}
index++;

// apply decimal where decimal not found
if(decimal < 100)
{
    if(decimal)
    {
        *number *= (100 / decimal);
    }
    else
    {
        *number *= 100;
    }
}

// make negative
if(negativeFlag)
    *number = *number * (-1);

return true;
}

inline bool copyInMoney32(const char * value, double *number)
{
    int      index      = 0;
    int      decimal     = 0;
    int      decimals    = 0;
    int      digitsAfterDec = 0;

    bool     negativeFlag = false;

    // convert each digit to a numeric portion
    while(value[index])
    {
        // handle $ . - All the rest assumed numeric
        switch(value[index])
        {
            case MONEY_SYMBOL:
                // ignore $ sign
                break;
            case NEGITIVE_SYMBOL:
                // set negative flag
                negativeFlag = true;
                break;
            case DECIMAL_SYMBOL:
                // set decimal
                decimal=1;
                decimals++;
                if(decimals > 1)
                    //more than 1 decimal point found
                    return false;
                break;
            default:
                // adjust decimal places
                decimal = decimal * 10;

```

```

// add digit to running total
if(value[index] >= '0' && value[index] <= '9')
{
    if(decimal)
        if(++digitsAfterDec > 2)
            return false;

    *number = (*number * 10) + (value[index] - '0');
}
else
{
    //non-numeric field inserted
    return false;
}
}
index++;

// apply decimal where decimal not found
if(decimal < 100)
{
    if(decimal)
    {
        *number *= (100 / decimal);
    }
    else
    {
        *number *= 100;
    }
}

// make negative
if(negativeFlag)
    *number = *number * (-1);

return true;
}

inline bool copyInMoney32(const char *value, float *number)
{
    int index = 0;
    int decimal = 0;
    int decimals = 0;
    int digitsAfterDec = 0;

    bool negativeFlag = false;

    // convert each digit to a numeric portion
    while(value[index])
    {
        // handle $ . - All the rest assumed numeric
        switch(value[index])
        {
            case MONEY_SYMBOL:
                // ignore $ sign
                break;

            case NEGATIVE_SYMBOL:
                // set negative flag
                negativeFlag = true;
                break;

            case DECIMAL_SYMBOL:
                // set decimal
                decimal=1;
                decimals++;
                if(decimals > 1)
                    //more than 1 decimal point found
                    return false;
                break;

            default:

```

```

// adjust decimal places
decimal = decimal * 10;

// add digit to running total
if(value[index] >= '0' && value[index] <= '9')
{
    if(decimal)
        if(++digitsAfterDec > 2)
            return false;

    *number = (*number * 10) + (value[index] - '0');
}
else
{
    //non-numeric field inserted
    return false;
}
}
index++;

// apply decimal where decimal not found
#ifdef DB2
if(decimal < 100)
{
    if(decimal)
    {
        *number *= (100 / decimal);
    }
    else
    {
        *number *= 100;
    }
}
#endif

// make negative
if(negativeFlag)
    *number = *number * (-1);

return true;
}

//
//
// copyInMoney
//
// Title : Convert char string money field to double
// Parameters : const char * - value string
// Return Value : double integer value
// Comments :
//
inline int copyInMoney(const char *value)
{
    char buf[20];
    int i,j,decimalFound,digitsAfterDecimal=0;

    int decimal=0;

    //walk past $ if present in char string
    if(*value == '$')
        *value++;

    int len=(int)strlen(value);
    for (i=0;i<len;i++)
    {
        if(value[i] == '.')
        {
            decimalFound++;
            if(decimalFound > 1)
                return -1;
        }
        if(value[i] == '-')

```

```

if (value[i] != '.')
{
    if(decimal)
    {
        if(digitsAfterDecimal<2)
            digitsAfterDecimal++;
        else
            return -1;
    }
    buf[i++] = value[i];
}
}
int amount = atoi(buf);

return amount;
}

//
// copyOutMoney64
//
// Title : Copy money data out of class array
// Parameters : char * - buffer to copy string 64 bit money into
//              INT64b - value
//              unsigned len - max number of bytes to copy
// Return Value : int - Length of copy
// Comments :
//
inline void copyOutMoney64(char *buffer,INT64b value,unsigned int len = DEFAULT_MONEY64_LEN)
{
    unsigned int index = len;

    int places = 0;

    bool negativeFlag = false;
    bool moneyFlag = true;

    // NULL terminate string
    buffer[index] = NULL;

    // check length > 0
    // if(index) return len;

    // handle negative value
    if(value < 0)
    {
        negativeFlag = true;
        value = value * (-1);
    }

    // break off each digit from value, fill if needed
    do
    {
        if(value)
        {
            // get next digit and add to buffer
            buffer[--index] = (char) (value % 10 + '0');
            value /= 10; places++;

            if(places == 2 && index)
            {
                places++;
                buffer[--index] = DECIMAL_SYMBOL;
            }
        }
        else
        {
            // add zeros to first place before decimal point on (i.e. 0.00)
            if(places < 2 || places == 3)
            {
                buffer[--index] = ZERO_SYMBOL;

```

```

}
else
{
    // add the decimal point
    if(places == 2)
    {
        buffer[--index] = DECIMAL_SYMBOL;
    }
    else
    {
        // add the negative indicator
        if(negativeFlag)
        {
            negativeFlag = false;
            buffer[--index] = NEGATIVE_SYMBOL;
        }
        else
        {
            // add the money indicator
            if(moneyFlag)
            {
                moneyFlag = false;
                buffer[--index] = MONEY_SYMBOL;
            }
            else buffer[--index] = NUMERIC_FILL;
        }
    }
}

// need to trace place for decimal point and zero fill
places++;
}
} while(index);
//return len;
}

//
// copyOutDateTime
//
// Title      : Copy date & time data out of class array
// Parameters  : char * - buffer to copy date & time string into
//              INT64b - value
// Return Value : int - Length of copy
// Comments    : Fixed length
//
inline int copyOutDateTime(char *buffer,INT64b value)
{
    datetime timestamp;

    // break value into time/date components
    calcOutDateTime(value,&timestamp);

    // put month into buffer
    *buffer++ = (char)((timestamp.month / 10) + '0');
    *buffer++ = (char)((timestamp.month % 10) + '0');
    *buffer++ = DATE_DELIMITER;

    // put day into buffer
    *buffer++ = (char)((timestamp.day / 10) + '0');
    *buffer++ = (char)((timestamp.day % 10) + '0');
    *buffer++ = DATE_DELIMITER;

    // put year into buffer
    int year = timestamp.year;
    *buffer++ = (char)((year / 1000) + '0');   year = year % 1000;
    *buffer++ = (char)((year / 100) + '0');   year = year % 100;
    *buffer++ = (char)((year / 10) + '0');
    *buffer++ = (char)((year % 10) + '0');
    *buffer++ = CHAR_FILL;

    // put hour into buffer
}

```

```

*buffer++ = (char)((timestamp.hour / 10) + '0');
*buffer++ = (char)((timestamp.hour % 10) + '0');
*buffer++ = TIME_DELIMITER;

// put minute into buffer
*buffer++ = (char)((timestamp.minute / 10) + '0');
*buffer++ = (char)((timestamp.minute % 10) + '0');
*buffer++ = TIME_DELIMITER;

// put second into buffer
*buffer++ = (char)((timestamp.second / 10) + '0');
*buffer++ = (char)((timestamp.second % 10) + '0');

*buffer = NULL;    return DEFAULT_DATETIME_LEN;
}

//
// copyOutTime
//
// Title      : Copy date data out of class array
// Parameters  : char * - buffer to copy date string into
//              INT64b - value
// Return Value : int - Length of copy
// Comments    : Fixed length
//
inline int copyOutDate(char *buffer,INT64b value)
{
    datetime timestamp;

    // break value into time/date components
    calcOutDate(value,&timestamp);

    // put month into buffer
    *buffer++ = (char)((timestamp.month / 10) + '0');
    *buffer++ = (char)((timestamp.month % 10) + '0');
    *buffer++ = DATE_DELIMITER;

    // put day into buffer
    *buffer++ = (char)((timestamp.day / 10) + '0');
    *buffer++ = (char)((timestamp.day % 10) + '0');
    *buffer++ = DATE_DELIMITER;

    // put year into buffer
    int year = timestamp.year;
    *buffer++ = (char)((year / 1000) + '0');   year = year % 1000;
    *buffer++ = (char)((year / 100) + '0');   year = year % 100;
    *buffer++ = (char)((year / 10) + '0');
    *buffer++ = (char)((year % 10) + '0');
    *buffer++ = CHAR_FILL;

    *buffer = NULL;

    return DEFAULT_DATE_LEN;
}

//
// copyOutTime
//
// Title      : Copy time data out of class array
// Parameters  : char * - buffer to copy time string into
//              INT64b - value
// Return Value : int - Length of copy
// Comments    : Fixed length TBD
//
inline int copyOutTime(char *buffer,INT64b value)
{
    datetime timestamp;

    // break value into time/date components
    calcOutDate(value,&timestamp);

    // put hour into buffer
}

```

```

*buffer++ = (char)((timestamp.hour / 10) + '0');
*buffer++ = (char)((timestamp.hour % 10) + '0');
*buffer++ = TIME_DELIMITER;

// put minute into buffer
*buffer++ = (char)((timestamp.minute / 10) + '0');
*buffer++ = (char)((timestamp.minute % 10) + '0');
*buffer++ = TIME_DELIMITER;

// put second into buffer
*buffer++ = (char)((timestamp.second / 10) + '0');
*buffer++ = (char)((timestamp.second % 10) + '0');

*buffer = NULL; return DEFAULT_TIME_LEN;
}

//
// copyOutDecimal64
//
// Title      : Copy decimal data out of class array
// Parameters  : char * - buffer to copy string 64 bit money into
//              INT64b - value
//              unsigned len - max number of bytes to copy
// Return Value : int - Length of copy
// Comments    :
//
inline int copyOutDecimal64(char *buffer,INT64b value,unsigned int len = DEFAULT_DECIMAL64_LEN)
{
    unsigned int index = len;

    int places = 0;

    bool negativeFlag = false;

    // NULL terminate string
    buffer[index] = NULL;

    // check length > 0
    if(index) return len;

    // handle negative value
    if(value < 0)
    {
        negativeFlag = true;
        value = value * (-1);
    }

    // break off each digit from value, fill if needed
    do
    {
        if(value)
        {
            // get next digit and add to buffer
            buffer[--index] = (char)(value % 10 + '0');
            value /= 10; places++;

            if(places == 2 && index)
            {
                places++;
                buffer[--index] = DECIMAL_SYMBOL;
            }
        }
        else
        {
            // add zeros to first place before decimal point on (i.e. 0.00)
            if(places < 2 || places == 3)
            {
                buffer[--index] = ZERO_SYMBOL;
            }
        }
        else
        {
            // add the decimal point
        }
    }
}

```

```

        if(places == 2)
        {
            buffer[--index] = DECIMAL_SYMBOL;
        }
        else
        {
            // add the negative indicator
            if(negativeFlag)
            {
                negativeFlag = false;
                buffer[--index] = NEGATIVE_SYMBOL;
            }
            else buffer[--index] = NUMERIC_FILL;
        }
    }

    // need to trace place for decimal point and zero fill
    places++;
} while(index);

return len;
}

// Macros
using namespace std;

#ifdef DEBUG
extern int debugFlag;
#else
extern int debugFlag;
#endif

inline BYTE8b *debugFileName(BYTE8b *filePath)
{
    BYTE8b *fileName = filePath + strlen(filePath);

    while(fileName != filePath)
    {
        if(*fileName == '/' || *fileName == '\\' && *(fileName + 1))
            return (fileName + 1);

        fileName--;
    }

    return filePath;
}

extern char *get_time_prefix(char *buffer);

#define DEBUGADDRESS(POINTER) hex << (void *) POINTER << dec

#define ERRORMSG(TEXT) {
    EnterCriticalSection(&errorMutex);

    char buff[50];

    errorStream << debugFileName(__FILE__)
    << "\n" << get_time_prefix(buff) << "\n" << __LINE__ << "\n"
    << _getpid() << "\n" << GetCurrentThreadId() << "\n"
    << TEXT;

    errorStream.flush();

    LeaveCriticalSection(&errorMutex);
}

#ifdef DEBUG

```

```

#define DEBUGMSG(TEXT) {
    EnterCriticalSection(&debugMutex);

    char buff[50];

    debugStream << debugFileName(__FILE__)
    << "\n" << get_time_prefix(buff) << "\n" << __LINE__ << "\n"
    << _getpid() << "\n" << GetCurrentThreadId() << "\n"
    << TEXT;

    debugStream.flush();

    LeaveCriticalSection(&debugMutex);
}

#define DEBUGSTRING(TEXT,LENGTH)
    debugVarString(TEXT,LENGTH)

#else
#define DEBUGMSG(TEXT);
#define DEBUGSTRING(TEXT,LENGTH);

#endif
#ifdef RT_DEBUG
#define RTMSG(TEXT) {
    EnterCriticalSection(&errorMutex);

    char buff[50];

    rStream << debugFileName(__FILE__)
    << "\n" << get_time_prefix(buff) << "\n" << __LINE__ << "\n"
    << _getpid() << "\n" << GetCurrentThreadId() << "\n"
    << TEXT;

    rStream.flush();

    LeaveCriticalSection(&errorMutex);
}

#endif
/* _COMMON_TPCC */

tpccIsapi/tpccIsapi.hpp

/*
*****
** Project      : AIX
** Component   : Performance/TPC-W Benchmark
** Name        : tpccIsapi.hpp
** Title       : ISAPI interface for tpcc
*****
** Copyright (c) 2001,2002 IBM Corporation
** All rights reserved
*****
** History     :
**   Developed at IBM Austin by the AIX RS/6000
**   performance group.
**
** Comments   :
*****
*/

```

```

#ifdef __tpccIsAPI_hpp__
#define __tpccIsAPI_hpp__

#include <windows.h>
#include <httplib.h>

#include "tpcc.h"
#include "htmlPhraser.h"

#include <iomanip>

#ifdef DB2
#include <db2tpcc.h>
#endif

#ifdef ORACLE
#include <oratpcc.h>
#endif

#include <comsvcs.h>

#ifdef RT_DEBUG
struct RT_DATA
{
    unsigned int count;
    unsigned int avg_rt[5][3];
    unsigned int min;
    unsigned int max;
};
#endif

// Terminal struct
// Terminal struct
struct TERM_ENTRY
{
    int terminalID;
    bool terminalInUse;
    int w_id;
    short d_id;
};

// COM interface
// COM interface
struct COM_HANDLE
{
    tpcc_com *comHandle;
    char *txnBuffer;
    int size;
};

// TXN handle
// TXN handle
struct TXN_HANDLE
{
    char htmlPage[MAX_HTML_PAGE_LEN];
    char htmlHeader[MAX_HTML_HEADER_LEN];
    char *urlString;
}

#ifdef RT_DEBUG
struct _timeb times[5];
#endif

//user data
int w_id;
int d_id;
int sync_id;
int term_id;
int conn_id;

COM_HANDLE comInterface;
};

```

```

////////////////////////////////////
// Definitions
////////////////////////////////////
#define INVALID_ITEM 100
#define HEADER "Content-Type:text/html\r\nContent-Length:
%dir\nConnection: Keep-Alive\r\n\r\n"
#define TLS_NULL 0xFFFFFFFF
#define ACCESS_TIMEOUT 3600000
//One hour in milli seconds

#define DELIVERY_LOG_SUCCESS_STR "--Tran %d Queue %d.%03d Start
%d.%03d\r\n_ID: %d CARRIER_ID: %d %s\r\nend-time: %d.%03d\r\n"

////////////////////////////////////
// Function Prototypes
////////////////////////////////////

int initDlvy();
int initTxnHandle(TXN_HANDLE *txnHandle);
int closeTxnHandle(TXN_HANDLE *txnHandle);
int readRegistryValues();
int getTerminal(int terminal, TXN_HANDLE *txnHandle);
int assignTerminal(TXN_HANDLE *txnHandle);
int getDBInstance();

void doHtml(TXN_HANDLE *txnHandle);
int doLoginForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doLoginResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doNewOrderForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doNewOrderResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doPaymentForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doPaymentResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doOrderStatusForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doOrderStatusResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doDeliveryForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doDeliveryResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doStockForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doStockResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);
int doExit(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);

int doLoginErrorPage(char *htmlPage, char *message);
int doNewOrderErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);
int doPaymentErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);
int doOrderStatusErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);
int doDeliveryErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);
int doStockErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle);

void dlvyThreadEntry(void *);
int queueDlvyTxn(int warehouse, short carrier_id);

int appendButtons(char *htmlPage);
int appendItems(char *htmlPage, short itemCount, short cmdID, start);
int appendHiddenFields(char *htmlPage, TXN_HANDLE *txnHandle);

int displayStatus(char *htmlPage, int rc);

#endif

tpccIsapi/tpccIsapi.cpp

/*
*****
** Project : AIX
** Component : Performance/TPC-C Benchmark
** Name : tpccIsapi.cpp
** Title : TPCC html processing

```

```

*****
** Copyright (c) 2003 IBM Corporation
** All rights reserved
*****
** History :
** Developed at IBM Austin by the AIX RS/6000
** performance group.
**
** Comments :
**
*****
*/

#include "stdafx.h"

#include "..\tpccCom\tpccCom.h"
#include "..\tpccCom\tpccCom_i.c"
#include "tpccIsapi.hpp"
#include <queue>

// For custom assert and trace handling with WebDbg.exe
[ module(name="tpccIsapi", type="dll") ];
[ emitidl(restricted) ];

#define _WIN32_DCOM

#ifdef _DEBUG
int debugFlag = 1;
#else
int debugFlag = 0;
#endif

////////////////////////////////////
// Globals
////////////////////////////////////

int maxDataSize; //max struct size of all txn(s)
int numUsers; //number of users that client will service.
int FirstClient;
int dlvyQueueLen; //static length of dlvy queue
int dlvyThreads; //number of dlvy threads to create
int dlvyBufferFreeSlots; //length of dlvy txn queue
int dlvyBufferSlotIndex; //index into next available slot in dlvy txn queue
int dlvyBufferThreadIndex; //thread index into dlvy txn queue
int nullDB; //null db on client(bypass com call).
int trace;

static DWORD threadLSIndex; //isapi thread local storage index
CRITICAL_SECTION isapiLock; //isapi lock
CRITICAL_SECTION errorLock; //error log file lock.
CRITICAL_SECTION termLock; //terminal array lock.
CRITICAL_SECTION dlvyQueueLock; //dlvy queue critical section lock
HANDLE dlvyThreadDone = INVALID_HANDLE_VALUE;
//dlvy thread exit event
HANDLE dlvyThreadSemaphore = INVALID_HANDLE_VALUE;
//dlvy thread wrk to do semaphore
int dlvyThreadID = 0;

//struct DLVYQUEUEUEDATA *dlvyQueue; //dlvy queue
HANDLE *dlvyThreadHandles; //ptr to array of thread
handles
queue<DLVYQUEUEUEDATA> *Dqueue;

TERM_ENTRY *termArray; //array of terminal
entries to store each users info.
int termNextFree; //next available slot in
terminal array

FILE *htmlDebug = NULL; //html debug file
FILE *errorLog = NULL; //error file
FILE *htmlTrace = NULL;

ofstream debugStream;

```

```

ofstream errorStream;
CRITICAL_SECTION debugMutex;
CRITICAL_SECTION errorMutex;

char dlvyLogPath[128] = {NULL};
char errorLogFile[128] = {NULL};
char htmlTraceLogFile[128] = {NULL};
char dbName[64] = {NULL};
char dbType[16] = {NULL};

typedef INT (*CONNECT_PTR)(char *dbName, void **connectHandle);
typedef INT (*DISCONNECT_PTR)(void *connectHandle);
typedef INT (*DLVY_FUNC_PTR)(dlvy_wrapper *dlvy, void *connectHandle);
typedef INT (*NORD_FUNC_PTR)(nord_wrapper *nord, void *connectHandle);
typedef INT (*PYMT_FUNC_PTR)(paym_wrapper *paym, void *connectHandle);
typedef INT (*ORDS_FUNC_PTR)(ords_wrapper *ords, void *connectHandle);
typedef INT (*STOK_FUNC_PTR)(stok_wrapper *stok, void *connectHandle);

HINSTANCE dbInstance;
CONNECT_PTR db_connect;
DISCONNECT_PTR db_disconnect;
DLVY_FUNC_PTR dlvyCall;

#ifdef RT_DEBUG
RT_DATA rt_data;
#ifdef WIN32
struct _timeb start_time; // time that the shared memory was initialized
#else
struct timeb start_time; // time that the shared memory was initialized
#endif
#endif

#ifdef RT_DEBUG
//
=====
UINT timestamp()
{
#ifdef WIN32
struct _timeb tb;
_timeb( &tb );
#else
struct timeb tb;
time( &tb );
#endif
// returns time since the SharedMem class was constructed
return ((tb.time - start_time.time) * 1000) + tb.millitm - start_time.millitm;
}
#endif

////////////////////////////////////
// Page functions arrays
////////////////////////////////////

typedef int (*pageFuncPtr) (htmlPhraser *commandBlock, TXN_HANDLE *txnHandle);

pageFuncPtr htmlPageFunctions[MAX_TRANSACTIONS] =
{
{doLoginForm},
{doNewOrderForm},
{doPaymentForm},
{doOrderStatusForm},
{doDeliveryForm},
{doStockForm},
{doExit},
{doLoginResults},
{doNewOrderResults},
{doPaymentResults},
{doOrderStatusResults},
{doDeliveryResults},
{doStockResults}
};

```

```

extern "C" DWORD WINAPI HttpExtensionProc(LPEXTENSION_CONTROL_BLOCK lpECB)
{
    struct TXN_HANDLE *txnHandle = NULL;

    txnHandle = (TXN_HANDLE *) TlsGetValue(threadLSIndex);

    if(txnHandle == NULL)
    {
        int rc = initTxnHandle(&txnHandle);
        if (rc != OK)
        {
            char response[256]; char htmlHeader[256];
            sprintf(response,"ERROR : Init txnHandle function failed.\n");

            size_t htmlPageLen = strlen(response);

            //add content length and keep alive header
            sprintf(htmlHeader,HEADER,htmlPageLen);
            lpECB->ServerSupportFunction(lpECB-
>ConnID,HSE_REQ_SEND_RESPONSE_HEADER,"200 OK",NULL,(DWORD*)htmlHeader);
            lpECB->WriteClient(lpECB->ConnID,response,(LPDWORD)&htmlPageLen,0);

            return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
        }

        txnHandle = (TXN_HANDLE *) TlsGetValue(threadLSIndex);
        if (txnHandle == NULL)
        {
            char response[256]; char htmlHeader[256];
            sprintf(response,"ERROR : Unable to retrieve txnHandle from TLS.\n");

            size_t htmlPageLen = strlen(response);

            //add content length and keep alive header
            sprintf(htmlHeader,HEADER,htmlPageLen);
            lpECB->ServerSupportFunction(lpECB-
>ConnID,HSE_REQ_SEND_RESPONSE_HEADER,"200 OK",NULL,(DWORD*)htmlHeader);
            lpECB->WriteClient(lpECB->ConnID,response,(LPDWORD)&htmlPageLen,0);

            return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
        }
    }
    try
    {
        txnHandle->urlString = (char*)lpECB->pszQueryString;

        DEBUGMSG("calling doHtml() w/ query string: "<< txnHandle->urlString << endl);
        doHtml(txnHandle);

        size_t htmlPageLen;
        htmlPageLen = strlen(txnHandle->htmlPage);
        if(htmlPageLen >= 4096)
        {
            ERRORMSG("WARNING: HTML PAGE IS >= 4096!, page
size:"<<htmlPageLen<<endl);
        }
        //add content length and keep alive header
        sprintf(txnHandle->htmlHeader,HEADER,htmlPageLen);
        size_t headerLen = strlen(txnHandle->htmlHeader);
        if(headerLen >= 256)
        {
            ERRORMSG("WARNING: HTML HEADER IS >= 256!, header
size:"<<headerLen<<endl);
        }

        //write response to user
        lpECB->ServerSupportFunction(lpECB-
>ConnID,HSE_REQ_SEND_RESPONSE_HEADER,"200 OK",NULL,(DWORD*)txnHandle->htmlHeader);
        lpECB->WriteClient(lpECB->ConnID,txnHandle-
>htmlPage,(LPDWORD)&htmlPageLen,0);
    }
}

```

```

        DEBUGMSG("HTML PAGE-->"<<endl<<txnHandle->htmlHeader<<txnHandle-
>htmlPage<<endl);
    }
    catch (...)
    {
        char response[256];
        ZeroMemory(response,256);
        char *ptr = response;

        appendText(&ptr,"<HTML><BODY> Error : Unhandled Exception </BODY></HTML>");
        DWORD cbResponse = sizeof(response)-1;

        //write response to user
        lpECB->ServerSupportFunction(lpECB-
>ConnID,HSE_REQ_SEND_RESPONSE_HEADER,"200 OK",NULL,(DWORD*)response);
        lpECB->WriteClient(lpECB->ConnID,response,&cbResponse,0);
    }

    return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

extern "C" BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO* pVer)
{
    // Create the extension version string, and copy string to HSE_VERSION_INFO structure.
    pVer->dwExtensionVersion = MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);

    // Copy description string into HSE_VERSION_INFO structure.
    strcpy(pVer->pszExtensionDesc, "TPCC ISAPI Extension");

    // Initialize isapi critical section
    InitializeCriticalSection(&isapiLock);

    // Initialize error log critical section
    InitializeCriticalSection(&errorLogLock);

    // Initialize terminal critical section
    InitializeCriticalSection(&termLock);

    // Initialize debug/error critical sections
    if(debugFlag)
        InitializeCriticalSection(&debugMutex);
    InitializeCriticalSection(&errorMutex);

    // Read registry values
    if(readRegistryValues() != OK)
        return(FALSE);

    // Initialize terminal array
    termArray = (TERM_ENTRY*) calloc(numUsers,sizeof(TERM_ENTRY));

/*
    if (errno == ENOMEM) {
        ERRORMSG("Error log file open."<<endl);
    }
*/

    termNextFree = 1;

    //open up error/debug streams
    errorStream.rdbuf( )->open(errorLogFile,ios::out);
    if(debugFlag)
        debugStream.rdbuf( )->open(htmlTraceLogFile,ios::out);

    ERRORMSG("Error log file open."<<endl);
#ifdef RT_DEBUG
    _ftime(&start_time); // get the time when shared memory is created
    ERRORMSG("start_time "<<start_time.time*1000+start_time.millitm<<endl);
#endif

    DEBUGMSG("Loading library for dlvy txn."<<endl);
    int rc = getDbInstance();
    if (rc != OK)
    {

```

```

        ERRORMSG("Error, unable to load database dll, rc:"<<rc);
        DEBUGMSG("Error, unable to load database dll, rc:"<<rc);

        return FALSE;
    }
    DEBUGMSG("Library loaded for dlvy txn."<<endl);

    DEBUGMSG("Calling initDlvy." <<endl);
    ERRORMSG("Calling initDlvy." <<endl);

    if(initDlvy() != OK)
        return (FALSE);

    DEBUGMSG("Initializing TLS." << endl);

    // Initialize thread local storage index
    threadLSIndex = TlsAlloc();
    if (threadLSIndex == TLS_NULL)
    {
        ERRORMSG("Isapi error: unable to initialize thread local storage(TLS), rc:" <<
GetLastError())<<endl);
        return(FALSE);
    }
    ERRORMSG("Initialized TLS." << endl);

    DEBUGMSG("sizeof out_neword_struct: "<<sizeof(struct out_neword_struct)<<endl);
    DEBUGMSG("sizeof in_neword_struct: "<<sizeof(struct in_neword_struct)<<endl);
    DEBUGMSG("sizeof out_payment_struct: "<<sizeof(struct out_payment_struct)<<endl);
    DEBUGMSG("sizeof in_payment_struct: "<<sizeof(struct in_payment_struct)<<endl);
    DEBUGMSG("sizeof out_ordstat_struct: "<<sizeof(struct out_ordstat_struct)<<endl);
    DEBUGMSG("sizeof in_ordstat_struct: "<<sizeof(struct in_ordstat_struct)<<endl);
    DEBUGMSG("sizeof out_delivery_struct: "<<sizeof(struct out_delivery_struct)<<endl);
    DEBUGMSG("sizeof in_delivery_struct: "<<sizeof(struct in_delivery_struct)<<endl);
    DEBUGMSG("sizeof out_stocklev_struct: "<<sizeof(struct out_stocklev_struct)<<endl);
    DEBUGMSG("sizeof in_stocklev_struct: "<<sizeof(struct in_stocklev_struct)<<endl);

    //compute the max struct size for com data construct
    maxDataSize = max(maxDataSize,sizeof(nord_wrapper));
    maxDataSize = max(maxDataSize,sizeof(paym_wrapper));
    maxDataSize = max(maxDataSize,sizeof(ords_wrapper));
    maxDataSize = max(maxDataSize,sizeof(dlvy_wrapper));
    maxDataSize = max(maxDataSize,sizeof(stok_wrapper));
    maxDataSize += 10;

    DEBUGMSG("max data struct size:"<<maxDataSize <<endl);
    ERRORMSG("max data struct size:"<<maxDataSize <<endl);

    return true;
}

extern "C" BOOL WINAPI TerminateExtension(DWORD dwFlags)
{
    // ERRORMSG("TerminateExtension"<<endl);
    delete Dqueue;
    return true;
}

/*
*****
** Name : initTxnHandle
** Description : Isapi thread initializes its own com interface structure.
** Parameters : TXN_HANDLE* isapi txn handle
** Returns : int - return code
** Comments :
**
*****
*/
int initTxnHandle(TXN_HANDLE **txnHandle)

```



```

{
    DEBUGMSG("Inside init txn handle, getting isapiLock." << endl);
    // ERRORMSG("Inside init txn handle, getting isapiLock." << endl);

    EnterCriticalSection(&isapiLock);

    HRESULT hres = NULL;

    try
    {
        DEBUGMSG("Got isapiLock, initializing txnHandle: "<<DEBUGADDRESS("txnHandle"<< endl);
        // ERRORMSG("Got isapiLock, initializing txnHandle: "<<DEBUGADDRESS("txnHandle"<< endl);

        *txnHandle = (TXN_HANDLE *) calloc(1, sizeof(TXN_HANDLE));
        if (*txnHandle == NULL)
        {
            ERRORMSG("Unable to allocated TXN_HANDLE, rc:"<<GetLastError()<<endl);
            return ERR;
        };

        (*txnHandle)->comInterface.comHandle = NULL;
        DEBUGMSG("Initializing txnHandle com data buffer to "<<maxDataSize<<"bytes"<<endl);
        ERRORMSG("Initializing txnHandle com data buffer to "<<maxDataSize<<"bytes"<<endl);
        (*txnHandle)->comInterface.txnBuffer = (char *) CoTaskMemAlloc(maxDataSize);
        if (!((*txnHandle)->comInterface.txnBuffer))
        {
            ERRORMSG("CoTaskMemAlloc() failed of size "<<maxDataSize<<"; rc:
"<<hres<<endl);
            return(ERR);
        };
        DEBUGMSG("txnHandle com data buffer initialized to " << maxDataSize << "bytes"
<<endl);
        // ERRORMSG("txnHandle com data buffer initialized to " << maxDataSize << "bytes"
<<endl);

        //((*txnHandle)->comInterface.comHandle = NULL;
        DEBUGMSG("Calling Colnitialize with txnHandle:
"<<DEBUGADDRESS("txnHandle"<<endl);
        // ERRORMSG("Calling Colnitialize with txnHandle:
"<<DEBUGADDRESS("txnHandle"<<endl);
        hres = ColnitializeEx(NULL, COINIT_MULTITHREADED);
        if (FAILED(hres))
        {
            ERRORMSG("ColnitializeEx() failed, rc: "<<hres<<endl);
            return(ERR);
        };

        struct _timeb          startTime;
        struct _timeb          endTime;

        DEBUGMSG("Calling CoCreateInstance with
txnHandle:"<<DEBUGADDRESS("txnHandle"<< endl);
        // ERRORMSG("Calling CoCreateInstance with
txnHandle:"<<DEBUGADDRESS("txnHandle"<< endl);
        _ftime(&startTime);
        hres =
CoCreateInstance(CLSID_tpcc_com, NULL, CLSCTX_SERVER, IID_itpcc_com, (void **)&(*txnHandle)-
>comInterface.comHandle);
        if (FAILED(hres))
        {
            _ftime(&endTime);
            //store error code in txnHandle
            ERRORMSG("CoCreateInstance() failed, code:"<<HRESULT_CODE(hres)<<"
facility:"<<HRESULT_FACILITY(hres)<<
            "hres:"<<hres<< " time waiting:"<<
            (((endTime.time - startTime.time)*1000)+
            (endTime.millitm - startTime.millitm))/1000.0)<<endl);

            DEBUGMSG("CoCreateInstance() failed, code:"<<HRESULT_CODE(hres)<<"
facility:"<<HRESULT_FACILITY(hres)<<
            "hres:"<<hres<< " time waiting:"<<
            (((endTime.time - startTime.time)*1000)+

```

```

            (endTime.millitm - startTime.millitm))/1000.0)<<endl);

            return(ERR);
        };

        _ftime(&endTime);
        DEBUGMSG("CoCreateInstance successful.txnHandle com initialized, time waiting for
object to be activated." <<
            (((endTime.time - startTime.time)*1000)+
            (endTime.millitm - startTime.millitm))/1000.0)<<endl);
        // ERRORMSG("CoCreateInstance successful.txnHandle com initialized, time waiting for
object to be activated." <<
        // (((endTime.time - startTime.time)*1000)+
        (endTime.millitm - startTime.millitm))/1000.0)<<endl);

        //call set complete to return object to pool.
        (*txnHandle)->comInterface.comHandle->doSetComplete();

        hres = (*txnHandle)->comInterface.comHandle->doDBInfo();
        (*txnHandle)->comInterface.comHandle->doSetComplete();

        //set the com buffers size
        DEBUGMSG("Setting txnHandle: " << DEBUGADDRESS("txnHandle") << "com buffer size
to " << maxDataSize<< endl);
        // ERRORMSG("Setting txnHandle: " << DEBUGADDRESS("txnHandle") << "com buffer size
to " << maxDataSize<< endl)
        (*txnHandle)->comInterface.size = maxDataSize;

        DEBUGMSG("txnHandle: "<<DEBUGADDRESS("txnHandle") <<"set to " << maxDataSize
<< endl);
        ERRORMSG("txnHandle: "<<DEBUGADDRESS("txnHandle") <<"set to " << maxDataSize
<< endl);

        TlsSetValue(threadLSIndex, *txnHandle);

        DEBUGMSG("txnHandle: "<<DEBUGADDRESS("txnHandle") <<"stored in TLS" << endl);
        ERRORMSG("txnHandle: "<<DEBUGADDRESS("txnHandle") <<"stored in TLS" << endl);

        ZeroMemory((*txnHandle)->htmlPage, MAX_HTML_PAGE_LEN);
        ZeroMemory((*txnHandle)->htmlHeader, MAX_HTML_HEADER_LEN);

        LeaveCriticalSection(&isapiLock);
        return(OK);
    }
    catch(...)
    {
        DEBUGMSG("Unhandled exeception in initTxnHandle, unlocking isapi lock" <<endl);
        ERRORMSG("Unhandled exeception in initTxnHandle, unlocking isapi lock" <<endl);
        LeaveCriticalSection(&isapiLock);
    };

    return ERR;
}

/*
*****
** Name          :          getDBInstance
** Description   :          load db specific lib based on dbType registry
**              :          value.
** Parameters   :
**              :
** Returns      :          int - return code
**              :          This function only exists for the dlvy threads
**              :          Dlvy threads hold direct connections to the database
**              :          and therefore need to know what db interface to talk to.
*****
*/
int getDBInstance()
{
    if(!nullDB)

```

```

{
    dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\nullDB.dll");
    if(dbInstance == NULL)
    {
        return ERR_NULL_DLL_NOT_LOADED;
    }
}
else if( (strcmp(dbType, "DB2") == 0) )
{
    dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccDB2glue.dll");
    if(dbInstance == NULL)
    {
        return ERR_DB2_DLL_NOT_LOADED;
    }
}
else if( (strcmp(dbType, "ORACLE") == 0) )
{
    ERRORMSG("Loading Oracle dll"<<endl);
    dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccOracleGlue.dll");
    if(dbInstance == NULL)
    {
        ERRORMSG("Could not Load Oracle dll"<<endl);
        return ERR_ORA_DLL_NOT_LOADED;
    }
    ERRORMSG("Loaded Oracle dll"<<endl);
}
else if( (strcmp(dbType, "SYBASE") == 0) )
{
    ERRORMSG("Loading Sybase dll"<<endl);
    dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccSybaseGlue.dll");
    if(dbInstance == NULL)
    {
        ERRORMSG("Could not Load Sybase dll"<<endl);
        return ERR_SYB_DLL_NOT_LOADED;
    }
    ERRORMSG("Loaded Sybase dll"<<endl);
}
else
{
    return ERR_UNKNOWN_DB;
}

ERRORMSG("Get address"<<endl);
db_connect = (CONNECT_PTR)GetProcAddress(dbInstance, "connect_db");
if(db_connect == NULL)
{
    ERRORMSG("Could not find connect_db function in dll"<<endl);
    return ERR_CONNECT_ADDRESS_NOT_FOUND;
}

ERRORMSG("Get address"<<endl);
dlvyCall = (DLVY_FUNC_PTR)GetProcAddress(dbInstance, "do_dlvy");
if(dlvyCall == NULL)
{
    ERRORMSG("Could not find do_dlvy in dll"<<endl);
    return ERR_DLVY_ADDRESS_NOT_FOUND;
}
}
return OK;
}

/*
*****
** Name          :          initDlvy
** Description   :          initialize dlvy threads/dlvy queueu
** Parameters   :
**              :
** Returns      :          int - return code
**              :
** Comments     :
*****
*/

```

```

int initDlvy()
{
    ERRORMSG(">>initDlvy"<<endl);
    ERRORMSG("dlvyQueueLen "<<dlvyQueueLen<<" dlvyThreads "<<dlvyThreads<<endl);
    // Initialize critical section
    InitializeCriticalSection(&dlvyQueueLock);

    //create dlvy queue
    dlvyQueue = (DLVYQUEUEDATA *) calloc(dlvyQueueLen,sizeof(DLVYQUEUEDATA));
    Dqueue = new queue<DLVYQUEUEDATA>;
    if (Dqueue == NULL)
    {
        ERRORMSG("calloc failed to allocate dlvyQueue"<<endl);
        return ERR_DLVY_QUEUE_CALLOC_FAIL;
    }
    ERRORMSG(">>calloc"<<endl);
    //init dlvy buffer critical section
    //InitializeCriticalSection(&dlvyQueueLock);

    dlvyThreadDone = CreateEvent(NULL,
                                TRUE,           //manual reset
                                FALSE,        //initially not signalled.
                                NULL);

    if(dlvyThreadDone == NULL)
    {
        DEBUGMSG("Error: dlvyThreadDone handled init failed,
GetLastError:"<<GetLastError()<<endl);

        ERRORMSG("Error : dlvyThreadDone handled init failed,
GetLastError:"<<GetLastError()<<endl);

        return ERR_DLVY_EVENT_INIT_FAILED;
    }
    //create dlvy semaphore
    dlvyThreadSemaphore = CreateSemaphore(NULL,0,dlvyQueueLen,NULL);
    if(dlvyThreadSemaphore == NULL)
    {
        DEBUGMSG("Error: dlvyThreadSemaphore semaphore init failed,
GetLastError:"<<GetLastError()<<endl);
        ERRORMSG("Error: dlvyThreadSemaphore semaphore init failed,
GetLastError:"<<GetLastError()<<endl);
        return ERR_DLVY_SEMAPHORE_INIT_FAILED;
    }

    //set number of free slots available in queue
    dlvyBufferFreeSlots = dlvyQueueLen;

    //index into next available slot in dlvy bxn queue
    dlvyBufferSlotIndex = 0;

    //thread index into dlvy bxn queue
    dlvyBufferThreadIndex = 0;

    dlvyThreadHandles = new HANDLE[dlvyThreads];

    //create threads
    for(int threadCount = 0,threadCount < dlvyThreads;threadCount++)
    {
        ERRORMSG(">>Calling dlvyThreadEntry"<<endl);
        dlvyThreadHandles[threadCount] = (HANDLE)_beginthread(dlvyThreadEntry,0,NULL);
        if(dlvyThreadHandles[threadCount] == INVALID_HANDLE_VALUE) {
            ERRORMSG(">>Calling dlvyThreadEntry failed"<<endl);
            return ERR_DLVY_THREAD_FAILED;
        }
    }

    return OK;
}
/*

```

```

*****
** Name          :      readRegistryValues
** Description   :      initialize isapi global variables from registry
** Parameters    :
** Returns       :      int - return code
** Comments     :
*****
*/
int readRegistryValues()
{
    HKEY registryKey;
    char value[MAX_STRING_LEN];
    DWORD regType;
    DWORD regValue;
    DWORD regValueSize = MAX_STRING_LEN;

    // ERRORMSG(">>readRegistryValues"<<endl);
    //open up registry key
    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTRY_SUB_KEY,0,KEY_READ,&registryKey) != ERROR_SUCCESS)
        return ERR_UNABLE_TO_OPEN_REG;

    //get null db flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;
    else
        nullDB = 0;

    //get num dlvy threads
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,DELIVERY_THREADS,0,&regType,(BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        dlvyThreads = regValue;
    else
        dlvyThreads = DEFAULT_DLVY_THREADS;

    //get dlvy queue len
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,DELIVERY_QUEUE_LEN,0,&regType,(BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        dlvyQueueLen = regValue;
    else
        dlvyQueueLen = DEFAULT_DLVY_QUEUE_LEN;

    //get the htmlTrace flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,HTML_TRACE,0,&regType,(BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        trace = regValue;
    else
        trace = 0;

    //get the client null db flag
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;
    else
        nullDB = 0;

    //get the num of users
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NUM_USERS,0,&regType,(BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        numUsers = regValue;
    else
        numUsers = DEFAULT_NUM_USERS;
}

```

```

//get dlvy log file path
regValueSize = sizeof(value);
if (RegQueryValueEx(registryKey,DELIVERY_LOG_PATH,0,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(dlvyLogPath,value);
else
    strcpy(dlvyLogPath,DEFAULT_DLVY_LOG_PATH);

//get global error log file path/name
regValueSize = sizeof(value);
if (RegQueryValueEx(registryKey,ERROR_LOG_FILE,0,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(errorLogFile,value);
else
    strcpy(errorLogFile,DEFAULT_ERROR_LOG_FILE);

//get global error log file path/name
regValueSize = sizeof(value);
if (RegQueryValueEx(registryKey,HTML_TRACE_LOG_FILE,0,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS )
    strcpy(htmlTraceLogFile,value);
else
    strcpy(htmlTraceLogFile,DEFAULT_HTML_TRACE_LOG_FILE);

//get db name
regValueSize = sizeof(value);
if (RegQueryValueEx(registryKey,DB_NAME,0,&regType,(BYTE *) &value,&regValueSize)== ERROR_SUCCESS )
    strcpy(dbName,value);
else
    strcpy(dbName,DEFAULT_DB_NAME);

//get db type
regValueSize = sizeof(value);
if (RegQueryValueEx(registryKey,DB_TYPE,0,&regType,(BYTE *) &value,&regValueSize)== ERROR_SUCCESS )
    strcpy(dbType,value);

//get First Client
regValueSize = sizeof(regValue);
if(RegQueryValueEx(registryKey,"FirstClient",0,&regType,(BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
    FirstClient = regValue;
else
    FirstClient = 1;

    RegCloseKey(registryKey);

    return OK;
}
/*
*****
** Name          :      doLoginForm
** Description   :      HTML Login page entry point
** Parameters    :      htmlPhraser  command block
                        char *      html result page
** Returns       :      int - return code
** Comments     :
*****
*/
int doLoginForm(htmlPhraser *commandBlock,TXN_HANDLE *txnHandle)
{
    char buffer[20];
    DEBUGMSG("Entering doLoginForm(),."<<endl);
    // ERRORMSG("Entering doLoginForm(),."<<endl);
    char *html=txnHandle->htmlPage;
}

```



```

int doNewOrderResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    DEBUGMSG("Entered doNewOrderResults" << endl);

    char *html=txnHandle->htmlPage;
    struct nord_wrapper *nord = NULL;

#ifdef RT_DEBUG
    _ftime(&txnHandle->times[1]);
#endif
    DEBUGMSG("Casting COM txnBuffer to nord struct" <<endl);
    nord = (nord_wrapper*)txnHandle->comInterface.txnBuffer;
    ZeroMemory(nord, sizeof(nord_wrapper));
    DEBUGMSG("COM txnBuffer initialized, validating input parameters" << endl);

    //set warehouse, district and customer id from command block
    nord->nord_s_w_ID = txnHandle->w_id;
    DEBUGMSG("nord w_id:" << nord->nord_s_w_ID << endl);

    if( (nord->nord_s_d_ID = atoi(commandBlock->get_D_ID())) == 0)
    {
        doNewOrderErrorPage(html_ERR_INVALID_D_ID,commandBlock,txnHandle);
        return OK;
    }
    DEBUGMSG("nord d_id:" << nord->nord_s_d_ID << endl);

    if( (nord->nord_s_c_ID = atoi(commandBlock->get_C_ID())) == 0)
    {
        doNewOrderErrorPage(html_ERR_INVALID_C_ID,commandBlock,txnHandle);
        return OK;
    }
    DEBUGMSG("nord c_id:" << nord->nord_s_c_ID << endl);

    int itemCmd          = ITEM_START;
    short itemComplete = 0;
    char field[256] = {NULL};

    for (int itemIndex=0; itemIndex<NORD_ITEMS; itemIndex++)
    {
        //supply warehouse
        if( *(commandBlock->get_ITEM_SUPP_W(itemIndex)) )

#ifdef DB2
        if ( (nord->nord_in_item[nord->nord_s_o_OL_CNT]s_OL_SUPPLY_W_ID =
        atoi(commandBlock->get_ITEM_SUPP_W(itemIndex))) == 0)
        #elif ORACLE
        if ( (nord->nord_s_OL_SUPPLY_W_ID[nord->nord_s_o_OL_CNT] =
        atoi(commandBlock->get_ITEM_SUPP_W(itemIndex))) == 0)
        #elif SYBASE
        if ( (nord->nord_s_OL_SUPPLY_W_ID[nord->nord_s_o_OL_CNT] =
        atoi(commandBlock->get_ITEM_SUPP_W(itemIndex))) == 0)
        #endif
        {
            doNewOrderErrorPage(html_ERR_INVALID_SUPPLY_W_ID,commandBlock,txnHandle);
            return OK;
        }
        else
            itemComplete++;

        //item number
        if( *(commandBlock->get_ITEM_ITEM_NUM(itemIndex)) )
        {
            if(itemComplete==1)
            {
#ifdef DB2
                if ( (nord->nord_in_item[nord->nord_s_o_OL_CNT]s_OL_I_ID =
                atoi(commandBlock->get_ITEM_ITEM_NUM(itemIndex))) == 0)
                #elif ORACLE
                if ( (nord->nord_s_OL_I_ID[nord->nord_s_o_OL_CNT] =
                atoi(commandBlock->get_ITEM_ITEM_NUM(itemIndex))) == 0)
                #elif SYBASE
                if ( (nord->nord_s_OL_I_ID[nord->nord_s_o_OL_CNT] =
                atoi(commandBlock->get_ITEM_ITEM_NUM(itemIndex))) == 0)
                #endif
                doNewOrderErrorPage(html_ERR_INVALID_ITEM_NUM,commandBlock,txnHandle);
                return OK;
            }
        }
    }

```

```

        if ( (nord->nord_s_OL_I_ID[nord->nord_s_o_OL_CNT] =
        atoi(commandBlock->get_ITEM_ITEM_NUM(itemIndex))) == 0)
        #endif
        {
            doNewOrderErrorPage(html_ERR_INVALID_ITEM_NUM,commandBlock,txnHandle);
            return OK;
        }
        else
            itemComplete++;
    }
    //missing previous value of item supp warehouse, flag error
    else
    {
        doNewOrderErrorPage(html_ERR_INVALID_SUPPLY_W_ID,commandBlock,txnHandle);
        return OK;
    }
}
else if( (itemComplete==1) ) //nothing in the command block, check to see if the previous
item value is present
{
    doNewOrderErrorPage(html_ERR_INVALID_ITEM_NUM,commandBlock,txnHandle);
    return OK;
}

//item qty
if( *(commandBlock->get_ITEM_QTY(itemIndex)) )
{
    if(itemComplete==2)
    {
#ifdef DB2
        if ( (nord->nord_in_item[nord-
>nord_s_o_OL_CNT]s_OL_QUANTITY = atoi(commandBlock->get_ITEM_QTY(itemIndex))) == 0)
        #elif ORACLE
        if ( (nord->nord_s_OL_QUANTITY[nord->nord_s_o_OL_CNT] =
        atoi(commandBlock->get_ITEM_QTY(itemIndex))) == 0)
        #elif SYBASE
        if ( (nord->nord_s_OL_QUANTITY[nord->nord_s_o_OL_CNT] =
        atoi(commandBlock->get_ITEM_QTY(itemIndex))) == 0)
        #endif
        {
            doNewOrderErrorPage(html_ERR_INVALID_ITEM_QTY,commandBlock,txnHandle);
            return OK;
        }
        else
            itemComplete++;
    }
    //missing previous value of item number
    else if( (itemComplete == 1)
    {
        doNewOrderErrorPage(html_ERR_INVALID_ITEM_NUM,commandBlock,txnHandle);
        return OK;
    }
    //missing 1st value of supp warehouse
    else
    {
        doNewOrderErrorPage(html_ERR_INVALID_SUPPLY_W_ID,commandBlock,txnHandle);
        return OK;
    }
}
else if(itemComplete==2) //nothing in the command block, check to see if
the previous item values are present
{
    doNewOrderErrorPage(html_ERR_INVALID_ITEM_NUM,commandBlock,txnHandle);
    return OK;
}
}
}

```

```

#ifdef DB2
    DEBUGMSG("nord item:" << nord->nord_s_o_OL_CNT << "SUPPLY_W_ID:" << nord-
>nord_s_OL_SUPPLY_W_ID[nord->nord_s_o_OL_CNT] <<
    "OL_I_ID:" << nord->nord_s_OL_I_ID[nord->nord_s_o_OL_CNT] << "
OL_QUANTITY:" << nord->nord_s_OL_QUANTITY[nord->nord_s_o_OL_CNT] <<endl);
#endif
    if(itemComplete == 3)
        nord->nord_s_o_OL_CNT++;

    itemComplete=0;

    DEBUGMSG("complete nord items:" <<nord->nord_s_o_OL_CNT<<" initializing remaini
ng unused items " << NORD_ITEMS - nord->nord_s_o_OL_CNT << " to 0" <<endl);
    for(int itemIndex=nord->nord_s_o_OL_CNT; itemIndex<NORD_ITEMS; itemIndex++)
    {
#ifdef DB2
        nord->nord_in_item[itemIndex]s_OL_SUPPLY_W_ID=0;
        nord->nord_in_item[itemIndex]s_OL_I_ID = 0;
        nord->nord_in_item[itemIndex]s_OL_QUANTITY = 0;
    #elif ORACLE
        nord->nord_s_OL_SUPPLY_W_ID[itemIndex]=0;
        nord->nord_s_OL_I_ID[itemIndex] = 0;
        nord->nord_s_OL_QUANTITY[itemIndex] = 0;
    #elif SYBASE
        nord->nord_s_OL_SUPPLY_W_ID[itemIndex]=0;
        nord->nord_s_OL_I_ID[itemIndex] = 0;
        nord->nord_s_OL_QUANTITY[itemIndex] = 0;
    #endif
    }

    DEBUGMSG("nord creating new order results html title page" <<endl);

    appendText(&html,"<HTML><HEAD><TITLE>TPC-C New Order Results</TITLE></HEAD>\n"
    "<BODY><FORM ACTION=\""
    APP_NAME
    "\" METHOD=\"GET\">\n");

    //append menu buttons
    html+=appendButtons(html);
    html+=appendHiddenFields(html,txnHandle);

    appendText(&html,"</FORM><CENTER><H3>New Order</H3><BR></CENTER>"
    "<PRE>"
    " 1 2 3 4 5 6 7 8
9\n"
//
//
"12345678901234567890123456789012345678901234567890123456789012345678901234
567890\n"
    "");

    //assume failure
    nord->out_nord_s_transtatus = -1;
    nord->nord_in_nord_len = sizeof(in_neword_struct);
    nord->out_nord_len = sizeof(out_neword_struct);

    DEBUGMSG("nord executing COM interface function," <<endl<<
    "nord c_id:" << nord->nord_s_c_ID << endl <<
    "nord w_id:" << nord->nord_s_w_ID << endl <<
    "nord d_id:" << nord->nord_s_d_ID << endl);
    HRESULT hres=0;
    if (txnHandle->comInterface.size > maxDataSize)
    {
        ERRORMSG("NO)txnHandle->comInterface.size " <<<txnHandle->comInterface.size);
    }
    try
    {
#ifdef RT_DEBUG
        _ftime(&txnHandle->times[2]);
#endif
        hres = txnHandle->comInterface.comHandle->doNewOrder(&txnHandle-
>comInterface.size, (UCHAR**) &txnHandle->comInterface.txnBuffer);
    #ifdef RT_DEBUG
        _ftime(&txnHandle->times[3]);
    #endif
}

```



```

html+=appendHiddenFields(html,txnHandle);

//int buffer for warehouse
char buffer[15];
//appendText(&html,"<PRE> 1 2 3 4 5 6 7 8 9\r\n"
"1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890\r\n"
"Warehouse: ");*/
appendText(&html,"<PRE>Warehouse: ");
appendText(&html,itoa(txnHandle->w_id,buffer,10),7,1);
appendText(&html,"District: <INPUT NAME=""
CMD_D_ID
"\" SIZE=1> Date:<BR>"
"Customer <INPUT NAME=""
CMD_C_ID
"\" SIZE=6> Name: Credit: %Disc.<BR>"
"Order Number: Number of Lines: W_tax:"
D_tax:<BR> <BR>"
// 1 2 3 4 5 6 7 8
9\r\n"
//1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890\r\n"
" Supp_W Item_Num Item_Name Qty Stock B/G
Price Amount <BR> ");
//append the 15 items commands
html+=appendItems(html,NORD_ITEMS,ITEM_START);
appendText(&html,message);
//seal up html page
appendText(&html,"<PRE><BODY></HTML>");
return OK;
}

/*
*****
** Name : doPaymentForm
** Description : HTML payment page entry point
** Parameters : htmlPhraser command block
char * html result page
** Returns :
** Comments : int - return code
**
*****
*/

int doPaymentForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
char *html=txnHandle->htmlPage;
#ifdef RT_DEBUG
ftime(&txnHandle->times[0]);
#endif
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Payment</TITLE></HEAD>\r\n"
"<BODY><FORM ACTION=""
APP_NAME
"\" METHOD=""GET"">\r\n"
"<CENTER><H3>Please Fill In Payment
Form.</H3></CENTER> <BR>\r\n"
"Submit Transaction <INPUT TYPE=""submit"" NAME=""
CMD_TXN_ID
"\" VALUE=""
CMD_PYMT
"\">");
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"<BR><PRE>\r\n"
>Date:<BR>"

```

```

"Warehouse: ");
char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));
appendSpaces(&html,10);
appendText(&html,"District: <INPUT NAME=""
CMD_D_ID
"\" SIZE=1>\r\n<BR>"
"<BR> <BR> <BR>"
"Customer:"
" <INPUT NAME=""
CMD_C_ID
"\" SIZE=5>"
" "
"Cust-Warehouse:"
" <INPUT NAME=""
CMD_C_W_ID
"\" SIZE=5>"
" "
"Cust-District:"
" <INPUT NAME=""
CMD_C_D_ID
"\" SIZE=1><BR>"
"Name: <INPUT NAME=""
CMD_C_NAME
"\" SIZE=20>");
appendText(&html,"
Since: <BR>"
" "
" Credit: <BR>"
" "
" %Disc: <BR>"
" "
"Amount Paid:"
" <INPUT NAME=""
CMD_AMT_PAID
"\" SIZE=10>"
" "
"New Cust-Balance:<BR>"
"Credit Limit:<BR> <BR>Cust-Data:<BR> <BR> <BR> <BR>"
</PRE>");
return OK;
}

/*
*****
** Name : doPaymentResults
** Description : HTML neworder page entry point
** Parameters : htmlPhraser command block
char * html result page
** Returns :
** Comments : int - return code
**
*****
*/

int doPaymentResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
char *html=txnHandle->htmlPage;
char buffer[50];
struct paym_wrapper *pymt = NULL;
#ifdef RT_DEBUG
ftime(&txnHandle->times[1]);
#endif
pymt = (paym_wrapper*)txnHandle->comInterface.txnBuffer;
ZeroMemory(pymt,sizeof(paym_wrapper));
//set login warehouse id from command block
pymt->in_paym.s_W_ID = txnHandle->w_id;

```

```

//set district from command block
#ifdef SYBASE
if( (pymt->in_paym.s_D_ID = (CS_TINYINT)atoi(commandBlock->get_D_ID())) == 0)
{
doPaymentErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandle);
return OK;
}
#else
if( (pymt->in_paym.s_D_ID = atoi(commandBlock->get_D_ID())) == 0)
{
doPaymentErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandle);
return OK;
}
#endif
//set customer id from command block
if( (pymt->in_paym.s_C_ID = atoi(commandBlock->get_C_ID())) == 0)
{
if( (commandBlock->get_C_NAME()) == NULL)
{
//no customer id nor customer last name specified.
doPaymentErrorPage(html,ERR_MISSING_C_ID_OR_CLAST,commandBlock,txnHandle);
return OK;
}
else
strcpy(pymt->in_paym.s_C_LAST,commandBlock->get_C_NAME());
}
else
{
//make sure that the user only inserted just c_id
if( (commandBlock->get_C_NAME()) != NULL)
{
doPaymentErrorPage(html,ERR_C_ID_OR_CLAST_ONLY,commandBlock,txnHandle);
return OK;
}
}
//get customer warehouse id field
if( (pymt->in_paym.s_C_W_ID = atoi(commandBlock->get_C_W_ID())) == 0)
{
doPaymentErrorPage(html,ERR_INVALID_C_W_ID,commandBlock,txnHandle);
return OK;
}
//get customer district id field
#ifdef SYBASE
if( (pymt->in_paym.s_C_D_ID = (CS_TINYINT)atoi(commandBlock->get_C_D_ID())) == 0)
{
doPaymentErrorPage(html,ERR_INVALID_C_D_ID,commandBlock,txnHandle);
return OK;
}
#else
if( (pymt->in_paym.s_C_D_ID = atoi(commandBlock->get_C_D_ID())) == 0)
{
doPaymentErrorPage(html,ERR_INVALID_C_D_ID,commandBlock,txnHandle);
return OK;
}
}
#endif
#ifdef DB2
pymt->in_paym.s_H_AMOUNT = (float)atof(commandBlock->get_AMT_PAID());
#else
if( (copyInMoney32(commandBlock->get_AMT_PAID()),&pymt->in_paym.s_H_AMOUNT))
{
doPaymentErrorPage(html,ERR_INVALID_PAYMENT_AMOUNT,commandBlock,txnHandle);
return OK;
}
}
#endif
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Payment Results</TITLE></HEAD>\r\n"
"<BODY><FORM ACTION=""
APP_NAME
"\" METHOD=""GET"">\r\n");

```

```

html+=appendButtons(html);

html+=appendHiddenFields(html,txnHandle);

appendText(&html,"<FORM><CENTER><H3>Payment</H3></CENTER>");

DEBUGMSG("pymt executing COM interface function,"<<endl<<
  "pymt c_id: " << pymt->in_paym.s_C_ID << endl <<
  "pymt w_id: " << pymt->in_paym.s_W_ID << endl <<
  "pymt d_id: " << pymt->in_paym.s_D_ID << endl);
//assume failure
pymt->out_paym.s_transtatus = -1;

DEBUGMSG("pym in,"<<endl<<
  "s_W_ID "<<pymt->in_paym.s_W_ID<<endl<<
  "s_C_W_ID "<<pymt->in_paym.s_C_W_ID<<endl<<
  "s_H_AMOUNT "<<pymt->in_paym.s_H_AMOUNT<<endl<<
  "s_D_ID "<<(int)pymt->in_paym.s_D_ID<<endl<<
  "s_C_D_ID "<<(int)pymt->in_paym.s_C_D_ID<<endl<<
  "s_C_ID "<<pymt->in_paym.s_C_ID<<endl<<
  "s_W_ID "<<pymt->in_paym.s_W_ID<<endl<<
  "s_C_LAST "<<pymt->in_paym.s_C_LAST<<endl);

HRESULT hres=0;
if (txnHandle->comInterface.size > maxDataSize)
{
  ERRORMSG("PY|txnHandle->comInterface.size "<<txnHandle->comInterface.size);
}
try
{
  #ifdef RT_DEBUG
    _ftime(&txnHandle->times[2]);
  #endif
  hres = txnHandle->comInterface.comHandle->doPayment(&txnHandle-
  >comInterface.size,(UCHAR*))&txnHandle->comInterface.txnBuffer);
  #ifdef RT_DEBUG
    _ftime(&txnHandle->times[3]);
  #endif
}
catch(...)
{
  html+=sprintf(html,"ERROR : Com Stock call caused exception to
  occur.</PRE></BODY></HTML>");
  ERRORMSG("COM+ exception [PY|txnHandle->comInterface.size "<<txnHandle-
  >comInterface.size);
  return OK;
}
if(FAILED(hres))
{
  html+=sprintf(html,"ERROR : pymt com call failed,
  rc:%x</PRE></BODY></HTML>",hres);
  ERRORMSG("ERROR : pymt com call failed, rc:"<<hres<<endl);
  DEBUGMSG("ERROR : pymt com call failed, rc:"<<hres<<endl);
  return OK;
}

DEBUGMSG("pymt executed OK,"<<endl<<
  "pymt c_id: " << pymt->in_paym.s_C_ID << endl <<
  "pymt w_id: " << pymt->in_paym.s_W_ID << endl <<
  "pymt d_id: " << pymt->in_paym.s_D_ID << endl);
txnHandle->comInterface.comHandle->doSetComplete();
pymt = (pymt_wrapper *)txnHandle->comInterface.txnBuffer;

//get return code
int rc = pymt->out_paym.s_transtatus;
if( rc != OK)
{
  html+=displayStatus(html,rc);
  appendText(&html,"</PRE></BODY></HTML>");
  return OK;
}
// appendText(&html,"<BR><PRE>\r\n");

```

```

// appendText(&html," 1 2 3 4 5 6 7 8<BR>");
//
appendText(&html,"1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890<BR>");

//start creating result body
appendText(&html,"<BR><PRE>\r\n"
  "Date: ");

#ifdef DB2
  appendText(&html,pymt->out_paym.s_H_DATE_time,18,1);
#else
  appendText(&html,pymt->out_paym.s_H_DATE_time,22,1);
#endif

appendText(&html,"<BR>"
  "Warehouse: ");
appendText(&html,itoa(pymt->in_paym.s_W_ID,buffer,10),6+24,1);
appendText(&html,"District: ");
appendText(&html,itoa(pymt->in_paym.s_D_ID,buffer,10),2,1);
appendText(&html,"<BR>");

//print out warehouse and district information
appendText(&html,pymt->out_paym.s_W_STREET_1,STREET_LEN+21,1);
appendText(&html,pymt->out_paym.s_D_STREET_1,STREET_LEN,1);
appendText(&html,"<BR>");

appendText(&html,pymt->out_paym.s_W_STREET_2,STREET_LEN+21,1);
appendText(&html,pymt->out_paym.s_D_STREET_2,STREET_LEN,1);
appendText(&html,"<BR>");

appendText(&html,pymt->out_paym.s_W_CITY,CITY_LEN+1,1);
appendText(&html,pymt->out_paym.s_W_STATE,STATE_LEN+1,1);
copyOutZip(buffer,pymt->out_paym.s_W_ZIP);
appendText(&html,buffer);

appendText(&html,pymt->out_paym.s_D_CITY,CITY_LEN+1,1);
appendText(&html,pymt->out_paym.s_D_STATE,STATE_LEN+1,1);
copyOutZip(buffer,pymt->out_paym.s_D_ZIP);
appendText(&html,buffer);

//print out customer information
appendText(&html,"<BR><BR>Customer: ");
appendText(&html,itoa(pymt->out_paym.s_C_ID,buffer,10),5+1,1);

appendText(&html,"Cust-Warehouse: ");
appendText(&html,itoa(pymt->in_paym.s_C_W_ID,buffer,10),6+1,1);

appendText(&html,"Cust-District: ");
appendText(&html,itoa(pymt->in_paym.s_C_D_ID,buffer,10));

//add customer information
appendText(&html,"<BR>Name: ");
appendText(&html,pymt->out_paym.s_C_FIRST,FIRST_NAME_LEN+1,1);
appendText(&html,pymt->out_paym.s_C_MIDDLE.INITIALS_LEN+1,1);
DEBUGMSG("Last name:"<<pymt->out_paym.s_C_LAST<<endl);
appendText(&html,pymt->out_paym.s_C_LAST,LAST_NAME_LEN+5,1);

appendText(&html,"Since: ");
#ifdef ORACLE
  appendText(&html,pymt->out_paym.c_since,d);
#elif DB2
  appendText(&html,pymt->out_paym.s_C_SINCE_time,18,1);
#elif SYBASE
  appendText(&html,pymt->out_paym.c_since);
#endif

appendText(&html,"<BR>");
appendSpaces(&html,8);

appendText(&html,pymt->out_paym.s_C_STREET_1,STREET_LEN+20,1);
appendText(&html,"Credit: ");
appendText(&html,pymt->out_paym.s_C_CREDIT);

appendText(&html,"<BR>");

```

```

appendSpaces(&html,8);

appendText(&html,pymt->out_paym.s_C_STREET_2,STREET_LEN+21,1);
appendText(&html,"%Disc: ");
html+=sprintf(html,"%2.2f",pymt->out_paym.s_C_DISCOUNT);

appendText(&html,"<BR>");
appendSpaces(&html,8);

appendText(&html,pymt->out_paym.s_C_CITY,CITY_LEN+1,1);

appendText(&html,pymt->out_paym.s_C_STATE,STATE_LEN+1,1);

copyOutZip(buffer,pymt->out_paym.s_C_ZIP);
appendText(&html,buffer,15,1);

appendText(&html,"Phone: ");
copyOutPhone(buffer,pymt->out_paym.s_C_PHONE);
appendText(&html,buffer);

appendText(&html,"<BR><BR>Amount Paid: $");
html+=sprintf(html,"%-9.2f",pymt->in_paym.s_H_AMOUNT);

appendText(&html,"New Cust-Balance: $");
html+=sprintf(html,"%-9.2f",pymt->out_paym.s_C_BALANCE);

appendText(&html,"<BR>Credit Limit: $");
html+=sprintf(html,"%-9.2f",pymt->out_paym.s_C_CREDIT_LIM);

appendText(&html,"<BR><BR>Cust-Data: ");
if(pymt->out_paym.s_C_CREDIT[0] == 'B' && pymt->out_paym.s_C_CREDIT[1] == 'C')
{
  appendCustData(&html,pymt->out_paym.s_C_DATA);
  appendText(&html,"<BR>");
}
else
  appendText(&html,"<BR><BR><BR>");

html+=displayStatus(html,rc);
appendText(&html,"</PRE></BODY></HTML>");

#ifdef RT_DEBUG
  _ftime(&txnHandle->times[4]);
  ERRORMSG(txnHandle->conn_id << "(1)-" << pymt->in_paym.s_W_ID << " " << pymt-
  >in_paym.s_D_ID << " " << pymt->out_paym.s_C_BALANCE << " " << txnHandle->times[0].time << " " <<
  >txnHandle->times[0].millitm << " " << txnHandle->times[1].time << " " << txnHandle->times[1].millitm << " " <<
  >txnHandle->times[2].time << " " << txnHandle->times[2].millitm << " " << txnHandle->times[3].time << " " <<
  >txnHandle->times[3].millitm << " " << txnHandle->times[4].time << " " << txnHandle-
  >times[4].millitm<<endl);
#endif
  return OK;
}
}

** Name : doPaymentErrorPage
** Description : append payment error body
** Parameters : char * html page result
char * error message
htmlPhraser * command block
** Returns : int - return code
** Comments :
*****
*/

int doPaymentErrorPage(char *htmlPage,char *message,htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)
{
  char *html=htmlPage;
  appendText(&html,"<HTML><HEAD><TITLE>TPC-C Payment</TITLE></HEAD>\r\n"

```

```

" <BODY><FORM ACTION=""
APP_NAME
"|" METHOD="GET"|>|<n"
"<CENTER><H3>Please Fill In Payment
Form.</H3></CENTER> <BR>|<n"
"Submit Transaction <INPUT TYPE="submit" NAME=""
CMD_TXN_ID
"|" VALUE=""
CMD_PYMT
"|">";
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"<BR><PRE>|<n"
"Date:<BR>*
"Warehouse:";
char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));
appendSpaces(&html,10);
appendText(&html,"District: <INPUT NAME=""
CMD_D_ID
"|" SIZE=1>|<n<BR>*
"<BR> <BR> <BR> <BR>
"Customer: "
"<INPUT NAME=""
CMD_C_ID
"|" SIZE=5>*
"
"Cust-Warehouse: "
"<INPUT NAME=""
CMD_C_W_ID
"|" SIZE=6>*
"
"Cust-District: "
"<INPUT NAME=""
CMD_C_D_ID
"|" SIZE=1><BR>*
"Name: <INPUT NAME=""
CMD_C_NAME
"|" SIZE=20>");
appendText(&html,"
Since: <BR> "
"
" Credit: <BR>"
"
" %Disc: <BR>"
"Amount Paid:
"<INPUT NAME=""
CMD_AMT_PAID
"|" SIZE=10>*
"
"New Cust-Balance:<BR>*
"Credit Limit:<BR> <BR> <BR> Cust-Data:<BR> <BR> <BR>
<BR>");
appendText(&html,message);
appendText(&html,"</PRE>");
return OK;
}
/*
*****
** Name : doOrderStatusForm
** Description :
** Parameters : HTML orderStatus page entry point
** Returns : htmlPhraser command block
char * html result page
** Comments : int - return code
**
*****
*/
int doOrderStatusForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
char *html=txnHandle->htmlPage;
#endif
_ftime(&txnHandle->times[0]);
#endif
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order Status</TITLE></HEAD>|<n"
"<BODY><FORM ACTION=""
APP_NAME
"|" METHOD="GET"|>|<n"
"<CENTER><H3>Please Fill In Order Status
Form.</H3></CENTER> <BR>|<n"
"Submit Transaction <INPUT TYPE="submit" NAME=""
CMD_TXN_ID
"|" VALUE=""
CMD_ORDS
"|">
"<BR>";
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"<PRE>|<n"
"Warehouse:";
char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));
appendText(&html,"
District: <INPUT NAME=""
CMD_D_ID
"|" SIZE=1>|<n<BR>"
"Customer: "
"<INPUT NAME=""
CMD_C_ID
"|" SIZE=5>*
"
"Name: "
"<INPUT NAME=""
CMD_C_NAME
"|" SIZE=20><BR>*
"Order-Number: Entry-Date: Carrier-
Number:<BR>*
"Supply-W Item-Num Qty Amount
Delivery:<BR></PRE>");
appendText(&html,"</BODY></HTML>");
return OK;
}
/*
*****
** Name : doOrderStatusResults
** Description :
** Parameters : HTML orderStatus page entry point
** Returns : htmlPhraser * command block
char * html result page
** Comments : int - return code
**
*****
*/
int doOrderStatusResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
char *html=txnHandle->htmlPage;
struct ords_wrapper *ords = NULL;
#endif
_ftime(&txnHandle->times[1]);
#endif
ords = (ords_wrapper *) txnHandle->comInterface.txnBuffer;
ZeroMemory(ords,sizeof(ords_wrapper));

```

```

//set warehouse login id from command blk
ords->in_ords.s_W_ID = txnHandle->w_id;
//set district login id from command blk
if( (ords->in_ords.s_D_ID = atoi(commandBlock->get_D_ID())) == 0)
{
doOrderStatusErrorPage(html,ERR_INVALID_D_ID,commandBlock,txnHandle);
return OK;
}
if( (ords->in_ords.s_C_ID = atoi(commandBlock->get_C_ID())) == 0)
{
if( (commandBlock->get_C_NAME()) == NULL)
{
//no customer id nor customer last name specified.
doOrderStatusErrorPage(html,ERR_MISSING_C_ID_OR_LAST,commandBlock,txnHandle);
return OK;
}
else
strcpy(ords->in_ords.s_C_LAST,commandBlock->get_C_NAME());
}
else
{
//make sure that the user only inserted just c_id
if( (commandBlock->get_C_NAME()) != NULL)
{
doOrderStatusErrorPage(html,ERR_C_ID_OR_LAST_ONLY,commandBlock,txnHandle);
return OK;
}
}
appendText(&html,"<HTML><HEAD><TITLE>TPC-C Order Status Results</TITLE></HEAD>|<n"
"<BODY><FORM ACTION=""
APP_NAME
"|" METHOD="GET"|>|<n");
html+=appendButtons(html);
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"</FORM>");
ords->out_ords.s_transtatus = -1;
DEBUGMSG("ords executing COM interface function,"<<endl<<
"ords c_id: " << ords->in_ords.s_C_ID << endl <<
"ords w_id: " << ords->in_ords.s_W_ID << endl <<
"ords d_id: " << ords->in_ords.s_D_ID << endl);
HRESULT hres=0;
if (txnHandle->comInterface.size > maxDataSize)
{
ERRORMSG("OS|txnHandle->comInterface.size "<<txnHandle->comInterface.size);
}
try
{
#endif
_ftime(&txnHandle->times[2]);
#endif
hres = txnHandle->comInterface.comHandle->doOrderStatus(&txnHandle-
>comInterface.size,(UCHAR*)&txnHandle->comInterface.txnBuffer);
#endif
_ftime(&txnHandle->times[3]);
#endif
}
catch(...)
{
html+=sprintf(html,"ERROR : ords com call caused
exception.</PRE></BODY></HTML>");
ERRORMSG("COM+ exception [OS|txnHandle->comInterface.size "<<txnHandle-
>comInterface.size);
return OK;
}

```





```

** Parameters :
**           htmlPhraser  command block
**           char*        html result page
** Returns    :          int - return code
** Comments   :
**
*****
*/
int doDeliveryForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=txnHandle->htmlPage;

#ifdef RT_DEBUG
    _ftime(&txnHandle->times[0]);
#endif
    appendText(&html, "<HTML><HEAD><TITLE>TPC-C Delivery</TITLE></HEAD>\r\n"
        "<BODY><FORM ACTION=\""
        APP_NAME
        "\" METHOD=\""GET\">\r\n"
        "<CENTER><H3>Delivery.</H3></CENTER>\r\n"
        "Submit Transaction <INPUT TYPE='submit' NAME=\""
        CMD_TXN_ID
        "\" VALUE=\""
        CMD_DLTV
        "\">");
    html+=appendHiddenFields(html,txnHandle);

    appendText(&html, "<BR> <PRE>"
        "Warehouse: ");
    char buffer[10];
    appendText(&html, itoa(txnHandle->w_id, buffer, 10));

    appendText(&html, "<BR> <BR>"
        "Carrier Number: "
        "<INPUT NAME=\""
        CMD_CARRIER_NUM
        "\" SIZE=1>"
        "</FORM></PRE>");

    appendText(&html, "</BODY></HTML>");

    return OK;
}
/*
*****
** Name      : doDeliveryResults
** Description : HTML payment page entry point
** Parameters : htmlPhraser* command block
**               TXN_HANDLE*  txn handle
** Returns    : int - return code
** Comments   :
**
*****
*/
int doDeliveryResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html = txnHandle->htmlPage;

    //declare delivery structure
    struct dlvy_wrapper dlvy;

#ifdef RT_DEBUG
    _ftime(&txnHandle->times[1]);
#endif
    //set warehouse login id from command blk
    dlvy.in_dlvy.s_W_ID = txnHandle->w_id;

```

```

//set the carrier id from command blk
if( dlvy.in_dlvy.s_O_CARRIER_ID = atoi(commandBlock->get_CARRIER_NUM()) == 0)
{
    doDeliveryErrorPage(html, ERR_INVALID_CARRIER, commandBlock, txnHandle);
    return OK;
}

//print title, add hidden fields , txn buttons
appendText(&html, "<HTML><HEAD><TITLE>TPC-C Delivery
Results</TITLE></HEAD>\r\n<BODY><FORM ACTION=\""
        APP_NAME
        "\" METHOD=\""GET\">\r\n");

html+=appendButtons(html);

html+=appendHiddenFields(html,txnHandle);

appendText(&html, "<FORM><CENTER><H3>Delivery</H3></CENTER>");

//call null db or com w/ delivery wrapper
#ifdef RT_DEBUG
    _ftime(&txnHandle->times[2]);
#endif
int rc = queueDlvyTxn(dlvy.in_dlvy.s_W_ID, dlvy.in_dlvy.s_O_CARRIER_ID);
#ifdef RT_DEBUG
    _ftime(&txnHandle->times[3]);
#endif
//if we are using the null db, rc will always be ok. The only time rc != OK is
//1) unable to queue txn because dlvy queue is full.
if( rc != OK)
{
    html+=displayStatus(html,rc);
    appendText(&html, "<PRE></BODY></HTML>\r\n");

    ERRORMSG("ERROR : Unable to queue dlvy txn, rc:"<<rc<<endl);
    return OK;
}

//start creating result body
appendText(&html, "Warehouse: ");

//get w_id from wrapper
char buffer[10];
appendText(&html, itoa(dlvy.in_dlvy.s_W_ID, buffer, 10));
appendText(&html, "<BR> <BR>Carrier Number: ");

//get carrier_id from wrapper
appendText(&html, itoa(dlvy.in_dlvy.s_O_CARRIER_ID, buffer, 10));
appendText(&html, "<BR> <BR>Execution Status: Delivery has been queued
</PRE></BODY></HTML>");

#ifdef RT_DEBUG
    _ftime(&txnHandle->times[4]);
    ERRORMSG(txnHandle->conn_id << "(4)-" << dlvy.in_dlvy.s_W_ID << " " <<
    dlvy.in_dlvy.s_O_CARRIER_ID << " " << txnHandle->times[0].time << " " << txnHandle->times[0].millitm
    << " " << txnHandle->times[1].time << " " << txnHandle->times[1].millitm << " " << txnHandle->times[2].time
    << " " << txnHandle->times[2].millitm << " " << txnHandle->times[3].time << " " << txnHandle-
    >times[3].millitm << " " << txnHandle->times[4].time << " " << txnHandle->times[4].millitm<<endl);
#endif
    return OK;
}
/*
*****
** Name      : doDeliveryErrorPage
** Description : HTML payment error page entry point
** Parameters :
**               char*        html result page
**               char*        error message
**               htmlPhraser  command block
**               TXN_HANDLE*  txn handle
**

```

```

** Returns    :          int - return code
** Comments   :
**
*****
*/
int doDeliveryErrorPage(char *htmlPage, char *message, htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=htmlPage;

    appendText(&html, "<HTML><HEAD><TITLE>TPC-C Delivery</TITLE></HEAD>\r\n"
        "<BODY><FORM ACTION=\""
        APP_NAME
        "\" METHOD=\""GET\">\r\n"
        "<CENTER><H3>Delivery.</H3></CENTER>\r\n"
        "Submit Transaction <INPUT TYPE='submit' NAME=\""
        CMD_TXN_ID
        "\" VALUE=\""
        CMD_DLTV
        "\">");
    html+=appendHiddenFields(html,txnHandle);

    appendText(&html, "<BR> <PRE>"
        "Warehouse: ");
    char buffer[15];
    appendText(&html, itoa(txnHandle->w_id, buffer, 10));

    appendText(&html, "<BR> <BR>"
        "Carrier Number: "
        "<INPUT NAME=\""
        CMD_CARRIER_NUM
        "\" SIZE=1> <BR>");

    appendText(&html, message);
    appendText(&html, "</PRE></BODY></HTML>");

    return OK;
}
/*
*****
** Name      : doStockForm
** Description : HTML stock page entry point
** Parameters : htmlPhraser  command block
**               TXN_HANDLE*  txn handle
** Returns    : int - return code
** Comments   :
**
*****
*/
int doStockForm(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=txnHandle->htmlPage;

#ifdef RT_DEBUG
    _ftime(&txnHandle->times[0]);
#endif
    appendText(&html, "<HTML><HEAD><TITLE>TPC-C Stock Level</TITLE></HEAD>\r\n"
        "<BODY><FORM ACTION=\""
        APP_NAME
        "\" METHOD=\""GET\">\r\n"
        "<CENTER><H3>Please Fill In Stock Form.</H3></CENTER>"
        "<BR>\r\n"
        "Submit Transaction <INPUT TYPE='submit' NAME=\""
        CMD_TXN_ID
        "\" VALUE=\""
        CMD_STOK

```

```

        ">");
html+=appendHiddenFields(html,txnHandle);

appendText(&html,"<PRE>"
           "Warehouse: ");

char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10),6+1,1);
appendText(&html,"District: ");

appendText(&html,itoa(txnHandle->d_id,buffer,10);
appendText(&html," <BR> <BR>"
           "Stock Level Threshold: "
           "<INPUT NAME='\"
           CMD_STK_THRESHOLD
           \"\" SIZE=1> <BR> <BR>"
           "Low Stock: <BR>"
           "</PRE>");

appendText(&html,"<FORM><BODY></HTML>");

return OK;
}

/*
*****
** Name           : doStockResults
** Description    :
**               HTML stock page entry point
** Parameters    :
**               htmlPhraser  command block
**               TXN_HANDLE *  handle for this transaction
** Returns       :
**               int - return code
** Comments      :
**
*****
*/

int doStockResults(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    char *html = txnHandle->htmlPage;

    struct stok_wrapper *stok;
#ifdef RT_DEBUG
        _ftime(&txnHandle->times[1]);
#endif

    stok = (stok_wrapper *)txnHandle->comInterface.txnBuffer;
    ZeroMemory(stok, sizeof(stok_wrapper));

    //set warehouse login id from command blk
    stok->in_stok_s_W_ID = txnHandle->w_id;

    //set district login id from command blk
    stok->in_stok_s_D_ID = txnHandle->d_id;

    //set stock level threshold id from command blk
    if( (stok->in_stok_s_threshold = atoi(commandBlock->get_STK_THRESHOLD())) == 0)
    {
        doStockErrorPage(html,ERR_INVALID_THRESHOLD,commandBlock,txnHandle);
        return OK;
    }

    //assume failure, set s_transtatus to err
    stok->out_stok_s_transtatus = INVALID_STATUS;

    //print title, add hidden fields , txn buttons
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock Level Results</TITLE></HEAD>\r\n"
              "<BODY><FORM ACTION='\"
              APP_NAME
              \"\" METHOD='\"GET\"'\>\r\n");

    html+=appendButtons(html);

    html+=appendHiddenFields(html,txnHandle);

```

```

appendText(&html,"</FORM>");

stok->out_stok_s_transtatus = OK;

DEBUGMSG("stok executing COM interface function,"<<endl<<
         "stok d_id: " << stok->in_stok_s_D_ID << endl <<
         "stok w_id: " << stok->in_stok_s_W_ID << endl <<
         "stok s_threshold: " << stok->in_stok_s_threshold << endl);

HRESULT hres=0;
if (txnHandle->comInterface.size > maxDataSize)
{
    ERRORMSG("[SR]txnHandle->comInterface.size "<<txnHandle->comInterface.size);
}
try
{
#ifdef RT_DEBUG
    _ftime(&txnHandle->times[2]);
#endif
    hres = txnHandle->comInterface.comHandle->doStockLevel(&txnHandle-
    >comInterface.size,(UCHAR*)&txnHandle->comInterface.txnBuffer);
#ifdef RT_DEBUG
    _ftime(&txnHandle->times[3]);
#endif
}
catch(...)
{
    html+=sprintf(html,"ERROR : Com Stock call caused exeception to
    occur.</PRE></BODY></HTML>");
    ERRORMSG("COM+ exeption [SR]txnHandle->comInterface.size "<<txnHandle-
    >comInterface.size);
    return OK;
}

//cast result back to stock structure
if(FAILED(hres))
{
    html+=sprintf(html,"ERROR : stok com call failed,
    rc:%ld</PRE></BODY></HTML>",hres);
    ERRORMSG("ERROR : stok com call failed, rc:"<<DEBUGADDRESS(hres)<<endl);
    DEBUGMSG("ERROR : stok com call failed, rc:"<<DEBUGADDRESS(hres)<<endl);
    return OK;
}

DEBUGMSG("stok executed OK,"<<endl<<
         "stok d_id: " << stok->in_stok_s_D_ID << endl <<
         "stok w_id: " << stok->in_stok_s_W_ID << endl <<
         "stok s_threshold: " << stok->in_stok_s_threshold << endl);

try
{
    txnHandle->comInterface.comHandle->doSetComplete();
}
catch(...)
{
    ERRORMSG("txnHandle address:"<<hex<<txnHandle<<
            "txnHandle->comInterface.comHandle:"<<hex<<txnHandle-
            >comInterface.comHandle<<
            "txnHandle->comInterface.txnBuffer:"<<hex<<(void *)txnHandle-
            >comInterface.txnBuffer<<endl);

    ERRORMSG("Com Stock setComplete call caused exeption to occur."<<endl);
    html+=sprintf(html,"ERROR : Com Stock setComplete call caused exeption to
    occur.</PRE></BODY></HTML>");
    return OK;
}

stok = (stok_wrapper *)txnHandle->comInterface.txnBuffer;
int rc = stok->out_stok_s_transtatus;
if(rc != OK)
{
    html+=displayStatus(html,rc);
}

```

```

        appendText(&html,"</PRE></BODY></HTML>");
        return OK;
    }

    //start creating result body
    appendText(&html,"<FORM><CENTER><H3>Stock Level</H3></CENTER>");
    appendText(&html,"<BR><PRE>\r\n"
              "Warehouse: ");

    //get w_id from wrapper
    char buffer[10];
    appendText(&html,itoa(stok->in_stok_s_W_ID,buffer,10),6+1,1);

    appendText(&html,"District: ");
    appendText(&html,itoa(stok->in_stok_s_D_ID,buffer,10);

    appendText(&html," <BR> <BR>"
              "Stock Level Threshold: ");
    appendText(&html,itoa(stok->in_stok_s_threshold,buffer,10);

    appendText(&html," <BR> <BR>"
              "Low Stock: ");
    appendText(&html,itoa(stok->out_stok_s_low_stock,buffer,10);
    appendText(&html," <BR> <BR>");

    html+=displayStatus(html,rc);
    appendText(&html,"</PRE></BODY></HTML>");

#ifdef RT_DEBUG
    _ftime(&txnHandle->times[4]);
    ERRORMSG(txnHandle->conn_id << "(3) " << stok->in_stok_s_W_ID << " " << stok-
    >in_stok_s_D_ID << " " << stok->out_stok_s_low_stock << " " << txnHandle->times[0].time << " " <<
    txnHandle->times[0].millitm << " " << txnHandle->times[1].time << " " << txnHandle->times[1].millitm << " " <<
    txnHandle->times[2].time << " " << txnHandle->times[2].millitm << " " << txnHandle->times[3].time << " " <<
    txnHandle->times[3].millitm << " " << txnHandle->times[4].time << " " << txnHandle-
    >times[4].millitm<<endl);
#endif

    return OK;
}

/*
*****
** Name           : doStockErrorPage
** Description    :
**               HTML stock page entry point
** Parameters    :
**               htmlPhraser  command block
**               char *        html result page
**               char *        query string
**               tpccHandle *  handle for this transaction
** Returns       :
**               int - return code
** Comments      :
**
*****
*/

int doStockErrorPage(char *htmlPage,char *message,htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    char *html=htmlPage;

    appendText(&html,"<HTML><HEAD><TITLE>TPC-C Stock Level</TITLE></HEAD>\r\n"
              "<BODY><FORM ACTION='\"
              APP_NAME
              \"\" METHOD='\"GET\"'\>\r\n"
              "<CENTER><H3>Please Fill In Stock Form.</H3></CENTER>"
              <BR>\r\n"
              "Submit Transaction <INPUT TYPE='\"submit\"\" NAME='\""
              CMD_TXN_ID
              \"\" VALUE='\""
              CMD_STOK

```

```

        ">");
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"<PRE>"
        "Warehouse: ");
char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10);
appendSpaces(&html,2);
appendText(&html,"District: ");
appendText(&html,commandBlock->get_D_ID());
appendText(&html," <BR> <BR>"
        "Stock Level Threshold: "
        "<INPUT NAME=''"
        "CMD_STK_THRESHOLD"
        "'\" SIZE=1> <BR> <BR>"
        "Low Stock: <BR>");
appendText(&html,message);
appendText(&html,"<PRE></FORM></BODY></HTML>");
return OK;
}

/*
*****
** Name : doExit
** Description : HTML exit page entry point
** Parameters : htmlPhraser command block
                char * html result page
** Returns : int - return code
** Comments :
*****
*/
int doExit(htmlPhraser *commandBlock, TXN_HANDLE *txnHandle)
{
    return (doLoginForm(commandBlock,txnHandle));
}

/*
*****
** Name : displayStatus
** Description : appends status string to the html page
** Parameters : char* html page
                int rc
** Returns : amount of characters the function appended
                to the html page
** Comments :
*****
*/
int displayStatus(char *htmlPage,int rc)
{
    char *html = htmlPage;
    appendText(&html,"");
    switch (rc)
    {
    case OK:
        appendText(&html,"Execution Status: Transaction Committed",50,1);
        break;
    case INVALID_ITEM:

```

```

        appendText(&html,"Execution Status: Item number is not valid",50,1);
        break;
    case INVALID_STATUS:
        appendText(&html,"Execution Status: ERROR: Rollback INVALID_STATUS",50,1);
        break;
    case INVALID_COM_STATUS:
        appendText(&html,"Execution Status: ERROR: Rollback COM FAILURE",50,1);
        break;
    case ERR_DLVY_QUEUE_FULL:
        appendText(&html,"Execution Status: ERROR: Rollback DLVY QUEUE FULL",50,1);
        break;
    default:
        appendText(&html,"Execution Status: ERROR: Rollback",50,1);
    };
    appendText(&html," ");
    return (int)(html - htmlPage);
}
/*
*****
** Name : appendButtons
** Description : append hidden field to recognize user after login
** Parameters : *htmlPage html result page
                *TXN_HANDLE txn handle
** Returns : int amount of characters the function
                appended to the html page
** Comments :
*****
*/
int appendHiddenFields(char *htmlPage, TXN_HANDLE *txnHandle)
{
    char *html = htmlPage;
    char buffer[15];
    appendText(&html,"<INPUT TYPE='hidden' NAME=''"
        "CMD_TERM_ID"
        "'\" VALUE=''"");
    appendText(&html,ittoa(txnHandle->term_id,buffer,10));
    appendText(&html,">\r\n");
    return (int)(html - htmlPage);
}
/*
*****
** Name : appendButtons
** Description : appends buttons transaction buttons to result page
** Parameters : *htmlPage
** Returns : amount of characters the function appended
                to the html page
** Comments :
*****
*/
int appendButtons(char *htmlPage)
{
    char *html = htmlPage;
    appendText(&html,"<INPUT TYPE='submit' NAME=''"
        "CMD_TXN_ID"
        "'\" VALUE=''"");

```

```

        CMD_NORD
        ">\r\n"
        "<INPUT TYPE='submit' NAME=''"
        "CMD_TXN_ID"
        "'\" VALUE=''"
        "CMD_PYMT"
        "'\" VALUE=''"
        ">\r\n"
        "<INPUT TYPE='submit' NAME=''"
        "CMD_TXN_ID"
        "'\" VALUE=''"
        "CMD_ORDS"
        "'\" VALUE=''"
        ">\r\n"
        "<INPUT TYPE='submit' NAME=''"
        "CMD_TXN_ID"
        "'\" VALUE=''"
        "CMD_DLVY"
        "'\" VALUE=''"
        ">\r\n"
        "<INPUT TYPE='submit' NAME=''"
        "CMD_TXN_ID"
        "'\" VALUE=''"
        "CMD_STOK"
        "'\" VALUE=''"
        ">\r\n"
        "<INPUT TYPE='submit' NAME=''"
        "CMD_TXN_ID"
        "'\" VALUE=''"
        "CMD_EXIT"
        "'\" VALUE=''"
        ">\r\n <BR>");
    return (int)(html - htmlPage);
}
/*
*****
** Name : appendItems
** Description : appends items to new order and order status page
** Parameters : *htmlPage html result page
                short items to append
                short item CMD id start
** Returns : amount of characters the function appended
                to the html page
** Comments :
*****
*/
int appendItems(char *htmlPage,short itemCount,short cmdIDStart)
{
    char *html = htmlPage;
    char numBuffer[MAX_INT_BUFFER];
    for(int item=0;item < itemCount;item++)
    {
        appendText(&html,"<BR> <INPUT NAME=''"");
        appendText(&html,ittoa(cmdIDStart++ ,numBuffer,10);
        appendText(&html,"\" SIZE=6> <INPUT NAME=''"");
        appendText(&html,ittoa(cmdIDStart++ ,numBuffer,10);
        appendText(&html,"\" SIZE=6> <INPUT NAME=''"");
        appendText(&html,ittoa(cmdIDStart++ ,numBuffer,10);
        appendText(&html,"\" SIZE=2>\r\n");
    }
    return (int)(html - htmlPage);
}
/*
*****
** Name : dlvyThreadEntry
** Description : dlvy thread worker entry point
** Parameters :

```

```

**
** Returns      :
**
** Comments    :
**              All dlvy threads created by initDlvy enter at
**              this point. They must first make a connection
**              to the database, then go to sleep.
**
**              Main isapi threads control dlvy worker semaphore
**              and signal when a dlvy txn is queued.
**
*****
*/

void dlvyThreadEntry(void *)
{
    int          rc = 0;

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " entered dlvyThreadEntry, calling
db_connect to db:" << dbName << endl);

    void *connectHandle;
    //connect to database.
    DEBUGMSG("ptr created, calling db_connect to db:" << dbName << endl);
    ERRORMSG("ptr created, calling db_connect to db:" << dbName << endl);
    rc = db_connect(dbName,&connectHandle);

    if(rc != OK)
    {
        ERRORMSG("dlvyThread " << GetCurrentThreadId() << " unable to connect to database,
rc:" << rc << endl);
        DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " unable to connect to database,
rc:" << rc << endl);
        return;
    }

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " connect to db:" << dbName << "
successful" << endl);

    ofstream dlvyStream;
    char logFileName[MAX_STRING_LEN] = {NULL};

    EnterCriticalSection(&isapiLock);
    //open dlvy log file for this thread
    sprintf(logFileName,"%s\\del_%d.txt",dlvyLogPath,dlvyThreadID);
    dlvyStream.rdbuf()->open(logFileName,ios::out);
    if(!dlvyStream.is_open())
    {
        ERRORMSG("dlvyThread " << GetCurrentThreadId() << " unable to open dlvy log "
<< dlvyLogPath << "\\del_" << dlvyThreadID << endl);
        DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " unable to open dlvy log "
<< dlvyLogPath << "\\del_" << dlvyThreadID << endl);
        return;
    }
    dlvyStream << setfill('0');
    //increment the global dlvy thread id
    dlvyThreadID++;

    LeaveCriticalSection(&isapiLock);

    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " dlvy log file name: " << logFileName << "
open." << endl);

    HANDLE      workerHandles[2];
    //handle array to store event to wait on

    struct DLVYQUEUEDATA      dlvyQueueData;          //dlvy queue struct to
store queued txn
    struct dlvy_wrapper      dlvyTxn;                //dlvy wrapper of db2
structs

    struct _timeb            endQueueTime;           //time stamp to queue
removal time

```

```

    struct _timeb            endProcessTime;         //time stamp for
end process time

    char          orderIDs[MAX_STRING_LEN*16] = {NULL}; //string to store oids for
each district
    int           bytesWritten = 0;
    int           dlvyCount    = 0;

    DEBUGMSG("dlvyThread entering work loop" << endl);

    //successful, while true
    while(true)
    {
        // try
        // {
        DEBUGMSG("dlvyThread initializing wait handles" << endl);

        //wait for both program exit AND if there is work to do
        workerHandles[0] = dlvyThreadDone;
        workerHandles[1] = dlvyThreadSemaphore;

        DEBUGMSG("dlvyThread going to sleep waiting for wrk" << endl);

        rc = WaitForMultipleObjects(2,&workerHandles[0],FALSE,INFINITE);

        DEBUGMSG("dlvyThread awake, checking wake condition" << endl);

        if(rc == WAIT_OBJECT_0)
            break;
        else if(rc == (WAIT_OBJECT_0+1))
        {
            DEBUGMSG("dlvyThread awake, wake condition of
dlvyThreadSemaphore" << endl);

            DEBUGMSG("dlvyThread trying to enter critical section" << endl);
            EnterCriticalSection(&dlvyQueueLock);

            DEBUGMSG("dlvyThread entered critical section" << endl);

            //remove queued dlvy txn
            dlvyQueueData = Dqueue->front();
            Dqueue->pop();

            DEBUGMSG("dlvyThread releasing critical section" << endl);
            LeaveCriticalSection(&dlvyQueueLock);

            //take enqueue time
            _ftime(&endQueueTime);

            DEBUGMSG("dlvyThread executing txn w_id:" << dlvyQueueData.warehouse
<< " carrier_id:" << dlvyQueueData.in_s_0_CARRIER_ID <<

            //prepare to call database
            dlvyTxn.in_dlvy_s_O_CARRIER_ID =
dlvyQueueData.in_s_0_CARRIER_ID;
            dlvyTxn.in_dlvy_s_W_ID        = dlvyQueueData.warehouse;

            #ifdef SYBASE
            dlvyTxn.in_dlvy_s_D_ID        = 1;
            #endif

            dlvyTxn.out_dlvy_s_transtatus = OK;

            //increment dlvy count
            dlvyCount++;

            DEBUGMSG("dlvyThread " << dlvyThreadID << " calling dlvy txn" << rc << endl);

            //call dlvy txn
            rc = dlvyCall(&dlvyTxn,connectHandle);

```

```

        _ftime(&endProcessTime);

        rc = dlvyTxn.out_dlvy_s_transtatus;

        #ifdef ORACLE
        DEBUGMSG("dlvy txn response time:" <<
(((endProcessTime.time - endQueueTime.time)*1000)+
(endProcessTime.millitm - endQueueTime.millitm))/1000.0 <<
"w_id:" << dlvyTxn.in_dlvy_s_W_ID << " carrier:"
<< dlvyTxn.in_dlvy_s_O_CARRIER_ID <<
" rc: " << rc << endl);
        #endif DB2
        DEBUGMSG("dlvy txn response time:" <<
(((endProcessTime.time - endQueueTime.time)*1000)+
(endProcessTime.millitm - endQueueTime.millitm))/1000.0 <<
"w_id:" << dlvyTxn.in_dlvy_s_W_ID << " carrier:"
<< dlvyTxn.in_dlvy_s_O_CARRIER_ID <<
" deadLocks:" << dlvyTxn.out_dlvy_deadlocks << " rc: " << rc
<< endl);
        #ifdef SYBASE
        DEBUGMSG("dlvy txn response time:" <<
(((endProcessTime.time - endQueueTime.time)*1000)+
(endProcessTime.millitm - endQueueTime.millitm))/1000.0 <<
"w_id:" << dlvyTxn.in_dlvy_s_W_ID << " carrier:"
<< dlvyTxn.in_dlvy_s_O_CARRIER_ID <<
" rc: " << rc << endl);
        #endif
        DEBUGMSG("dlvyThread dlvy s_transtatus:" << rc << endl);

        if(rc == OK)
        {
            bytesWritten=0;
            char *buffer = orderIDs;

            #ifdef SYBASE
            for(int districtIndex=1,districtIndex <=
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            {
                if(dlvyTxn.out_dlvy_s_O_ID[districtIndex] == 0)
                    bytesWritten = sprintf(buffer,"nD_ID %d had no new
orders",districtIndex);
                else
                    bytesWritten = sprintf(buffer,"%d
",dlvyTxn.out_dlvy_s_O_ID[districtIndex]);

                buffer+=bytesWritten;
            }
            #else
            for(int districtIndex=0,districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            {
                if(dlvyTxn.out_dlvy_s_O_ID[districtIndex] == 0)
                    bytesWritten = sprintf(buffer,"nD_ID %d had no new
orders",districtIndex);
                else
                    bytesWritten = sprintf(buffer,"%d
",dlvyTxn.out_dlvy_s_O_ID[districtIndex]);

                buffer+=bytesWritten;
            }
            #endif
            }
            else
                sprintf(orderIDs,"nDelivery transaction failed");

            dlvyStream << "--Tran " << dlvyCount <<
" Queue
"<< dlvyQueueData.enqueueTime.time << " " << setw(3) <<
dlvyQueueData.enqueueTime.millitm <<

```

```

        " Start "<<endQueueTime.time<<"<< setw(3)
<<
        endQueueTime.millitm<<endl<<
        "W_ID:"<<dlvyQueueData.warehouse<<" "<<
        " CARRIER_ID:
" "<<orderIDs<<endl<<
        "end-time:"<<endProcessTime.time<<"<<

setw(3) <<
        endProcessTime.millitm<<endl;

        dlvyStream.flush();
        DEBUGMSG("dlvyThread All done" << endl);
    }
    catch(...)
    {
        ERRORMSG("ERROR : Unhandled exeception in dlvy thread. Thread
        exiting"<<endl);
        fprintf(dlvyLog,"ERROR : Unhandled exeception in dlvy thread %ld. Thread
        exiting.\n",GetCurrentThreadId());
        fflush(dlvyLog);

        LeaveCriticalSection(&dlvyQueueLock);
        throw;
    }
} //end while true
}

/*
*****
** Name          : queueDlvyTxn
** Description   : function queues dlvy txn in dlvy queue
** Parameters    :
**                int      warehouse
**                short    carrier
** Returns       :
**                int      error code
** Comments      :
**                Function will queue dlvy txn if 2 points are true
**                1) We have room in our dlvy buffer
**                2) We writing over the end of the queue
*****
*/

int queueDlvyTxn(int warehouse, short carrier_id)
{
    struct DLVYQUEUEDATA    dlvyQueueData;           //dlvy queue struct to
    store queued txn

    DEBUGMSG("Taking lock to queue dlvy txn.");

    dlvyQueueData.warehouse = warehouse;
    dlvyQueueData.in_s_0_CARRIER_ID = carrier_id;

    _ftime(&dlvyQueueData.enqueueTime);
    EnterCriticalSection(&dlvyQueueLock);

    DEBUGMSG("Lock aquired to queue dlvy txn");
    Dqueue->push(dlvyQueueData);

    LeaveCriticalSection(&dlvyQueueLock);

    //release semaphore to wake thread that there is work
    ReleaseSemaphore(dlvyThreadSemaphore,1,NULL);

    return OK;
}

/*
*****
** Name          : doHtml

```

```

** Description   : HTML processing page entry point
** Parameters    : txn handle
** Returns       : int - return code
** Comments      :
*****
*/

void doHtml(TXN_HANDLE *txnHandle)
{
    DEBUGMSG("Entered doHtml(), parsing query string:"<< txnHandle->urlString <<" into command
    block"<<endl);
    // ERRORMSG("Entered doHtml(), parsing query string:"<< txnHandle->urlString <<" into command
    block"<<endl);
    htmlPhraser    commandBlock(txnHandle->urlString);
    DEBUGMSG("Query string parsed. command:"<<commandBlock.getCommandId() <<" user's
    terminal id." <<commandBlock.get_TERM_ID() <<endl);

    int terminalID = atoi(commandBlock.get_TERM_ID());
    int commandID = commandBlock.getCommandId();

    DEBUGMSG("User sent in a terimal id:"<<terminalID<<" , checking to see if user has logged in
    before"<<endl);
    if(terminalID > 0)
    {
        DEBUGMSG("Terminal id > 0, user has logged in already, terminalID:"<<terminalID<<"
        retrieving warehouse district pair"<<endl);
        if(getTerminal(terminalID,txnHandle) != OK)
            return;
        DEBUGMSG("User had valid terminal id, user's login warehouse:"<<txnHandle->w_id<<"
        district:"<<txnHandle->d_id<<endl);
    }
    else
    {
        DEBUGMSG("User did not submit a terminal id or valid terminal id, ensure that the user is
        trying to log in."<<endl);
        if( (commandID != TXN_LOGIN) && (commandID != TXN_LOGIN_RESULTS) )
        {
            DEBUGMSG("ERROR : User has not logged in."<<endl);
            ERRORMSG("ERROR : User has not logged in."<<endl);
            sprintf(txnHandle->htmlPage,"ERROR : User has not logged in or did not submit a
            valid terminal.");
            return;
        }
        DEBUGMSG("User is in process of logging in, commandID:"<<commandID<<endl);
    }

    DEBUGMSG("Calling html page function:"<<commandBlock.getCommandId()<<endl);
    int rc = htmlPageFunctions[commandBlock.getCommandId()](&commandBlock,txnHandle);
    DEBUGMSG("Return from html page function:"<<commandBlock.getCommandId()<<endl);

    return;
}

/*
*****
** Name          : getTerminal
** Description   : retrieves terminal information based on terminal id
** Parameters    :
**                int      terminal id
**                TERM_HANDLE*  txn handle
** Returns       :
**                int - return code
** Comments      :
*****
*/

int getTerminal(int terminal,TXN_HANDLE *txnHandle)
{

```

```

// ERRORMSG(">>getTerminal"<<endl);
//check to see if terminal id is out of range
if(terminal >= numUsers)
{
    //terminal id not valid.
    sprintf(txnHandle->htmlPage,"ERROR : Client does not support more than %d users,
    terminal id:%d",numUsers,terminal);
    ERRORMSG("ERROR : Client does not support more than "<<numUsers<<" users,
    terminal id:"<<terminal<<endl);
    return ERR;
}

//check if terminal id is points to a not in use terminal
if(!((termArray+terminal)->terminalInUse)
{
    sprintf(txnHandle->htmlPage,"ERROR : Terminal id given points to a not in use terminal,
    terminal=%d.",terminal);
    ERRORMSG("ERROR : Terminal id given points to a not in use terminal,
    terminal="<<terminal<<endl);
    return ERR;
}

DEBUGMSG("Storing terminal warehouse, district , and initial term id for user:"<<terminal<<endl);

//assign terminal values to txn_handle
txnHandle->d_id = termArray[terminal].d_id;
txnHandle->w_id = termArray[terminal].w_id;
txnHandle->term_id = termArray[terminal].terminalID;

DEBUGMSG("Users terminal:"<<terminal<<" , stored warehouse:"<<txnHandle->w_id<<
" district:"<<txnHandle->d_id<<" terminalID stored:"<<txnHandle->term_id<<endl);

return OK;
}

/*
*****
** Name          : assignTerminal
** Description   : assigns terminal index to user
** Parameters    : TERM_HANDLE*  txn handle
** Returns       : int - return code
** Comments      :
*****
*/

int assignTerminal(TXN_HANDLE *txnHandle)
{
// ERRORMSG(">>assignTerminal"<<endl);
EnterCriticalSection(&termLock);

//check if terminal array is full.
if(termNextFree == numUsers)
{
    LeaveCriticalSection(&termLock);
    return ERR;
}

DEBUGMSG("Storing user warehouse:"<<txnHandle->w_id<<" district:"<< txnHandle->d_id<<
" in terminal slot:"<<termNextFree<<endl);

//store users w_id and d_id
termArray[termNextFree].d_id = txnHandle->d_id;
termArray[termNextFree].w_id = txnHandle->w_id;

//set terminal slot to be in use
termArray[termNextFree].terminalInUse = true;
termArray[termNextFree].terminalID = termNextFree;
//in txn handle, set the terminal id
txnHandle->term_id = termNextFree;

```

```

//increment to next free terminal.
termNextFree++;

DEBUGMSG("User warehouse:<<txnHandle->w_id<<" district:<< txnHandle->d_id <<
" stored in terminal slot:<<txnHandle->term_id<<" next terminal
free:<<termNextFree<<endl);

LeaveCriticalSection(&termLock);

return OK;
}

```

## tpccIsapi/tpccIsapi.sln

```

Microsoft Visual Studio Solution File, Format Version 10.00
# Visual Studio 2008
Project("8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942") = "tpccCom", "..\tpccCom\tpccCom.vcproj",
"(3401AE40-CD75-4B1B-8302-6A30DFD4B3B7)"
EndProject
Project("8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942") = "tpccIsapi", "tpccIsapi.vcproj", "(619C38BD-
A8B2-433B-9D21-9A4A64F5C6BD)"
EndProject
Project("8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942") = "tpccDB2glue",
"..\tpccDB2glue\tpccDB2glue.vcproj", "(00B934F6-BE49-4022-AF4A-4A785DF42AEE)"
EndProject
Global
GlobalSection(SolutionConfigurationPlatforms) = preSolution
Debug|Win32 = Debug|Win32
Release|Win32 = Release|Win32
EndGlobalSection
GlobalSection(ProjectConfigurationPlatforms) = postSolution
(3401AE40-CD75-4B1B-8302-6A30DFD4B3B7).Debug|Win32.ActiveCfg = Debug|Win32
(3401AE40-CD75-4B1B-8302-6A30DFD4B3B7).Debug|Win32.Build.0 = Debug|Win32
(3401AE40-CD75-4B1B-8302-6A30DFD4B3B7).Release|Win32.ActiveCfg =
Release|Win32
(3401AE40-CD75-4B1B-8302-6A30DFD4B3B7).Release|Win32.Build.0 = Release|Win32
(619C38BD-A8B2-433B-9D21-9A4A64F5C6BD).Debug|Win32.ActiveCfg = Debug|Win32
(619C38BD-A8B2-433B-9D21-9A4A64F5C6BD).Debug|Win32.Build.0 = Debug|Win32
(619C38BD-A8B2-433B-9D21-9A4A64F5C6BD).Release|Win32.ActiveCfg =
Release|Win32
(619C38BD-A8B2-433B-9D21-9A4A64F5C6BD).Release|Win32.Build.0 = Release|Win32
(00B934F6-BE49-4022-AF4A-4A785DF42AEE).Debug|Win32.ActiveCfg = Debug|Win32
(00B934F6-BE49-4022-AF4A-4A785DF42AEE).Debug|Win32.Build.0 = Debug|Win32
(00B934F6-BE49-4022-AF4A-4A785DF42AEE).Release|Win32.ActiveCfg =
Release|Win32
(00B934F6-BE49-4022-AF4A-4A785DF42AEE).Release|Win32.Build.0 = Release|Win32
(29744379-4E5D-4DA8-98BD-B3D9B35209D9).Debug|Win32.ActiveCfg = Debug|Win32
(29744379-4E5D-4DA8-98BD-B3D9B35209D9).Debug|Win32.Build.0 = Debug|Win32
(29744379-4E5D-4DA8-98BD-B3D9B35209D9).Release|Win32.ActiveCfg =
Release|Win32
(29744379-4E5D-4DA8-98BD-B3D9B35209D9).Release|Win32.Build.0 = Release|Win32
(6A69ADF3-B73C-4912-8071-43FF2C3ACAAE).Debug|Win32.ActiveCfg = Debug|Win32
(6A69ADF3-B73C-4912-8071-43FF2C3ACAAE).Debug|Win32.Build.0 = Debug|Win32
(6A69ADF3-B73C-4912-8071-43FF2C3ACAAE).Release|Win32.ActiveCfg =
Release|Win32
(6A69ADF3-B73C-4912-8071-43FF2C3ACAAE).Release|Win32.Build.0 = Release|Win32
(2DB5D990-9180-4DD7-AFE0-6837D2A81D4F).Debug|Win32.ActiveCfg = Debug|Win32
(2DB5D990-9180-4DD7-AFE0-6837D2A81D4F).Debug|Win32.Build.0 = Debug|Win32
(2DB5D990-9180-4DD7-AFE0-6837D2A81D4F).Release|Win32.ActiveCfg =
Release|Win32
(32B36E42-FBDC-4A14-A8CC-3F96BA0EA9BA).Debug|Win32.ActiveCfg = Debug|Win32
(32B36E42-FBDC-4A14-A8CC-3F96BA0EA9BA).Debug|Win32.Build.0 = Debug|Win32
(32B36E42-FBDC-4A14-A8CC-3F96BA0EA9BA).Release|Win32.ActiveCfg =
Release|Win32
(32B36E42-FBDC-4A14-A8CC-3F96BA0EA9BA).Release|Win32.Build.0 =
Release|Win32
(4A431713-96DF-4ABE-B422-89FDDEBCC2E8).Debug|Win32.ActiveCfg = Debug|Win32
(4A431713-96DF-4ABE-B422-89FDDEBCC2E8).Release|Win32.ActiveCfg =
Release|Win32
(2117DDAD-8B12-4EA1-B188-91B66EF10D7D).Debug|Win32.ActiveCfg = Debug|Win32
(2117DDAD-8B12-4EA1-B188-91B66EF10D7D).Debug|Win32.Build.0 = Debug|Win32

```

```

(2117DDAD-8B12-4EA1-B188-91B66EF10D7D).Release|Win32.ActiveCfg =
Release|Win32
(2117DDAD-8B12-4EA1-B188-91B66EF10D7D).Release|Win32.Build.0 = Release|Win32
EndGlobalSection
GlobalSection(SolutionProperties) = preSolution
HideSolutionNode = FALSE
EndGlobalSection
EndGlobal

```

## tpccIsapi/tpccIsapi.def

: tpccIsapi.def : declares the module parameters for the DLL.

```
LIBRARY "tpccIsapi"
```

```
EXPORTS
HttpExtensionProc
GetExtensionVersion
TerminateExtension

```

## tpccIsapi/tpccIsapi.rc

// Microsoft Visual C++ generated resource script.

```
//
#include "resource.h"
```

```
#define APSTUDIO_READONLY_SYMBOLS
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "winres.h"
#include "atlsrvres.h"
```

```
////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS
//
//def APSTUDIO_INVOKED
//
// TEXTINCLUDE
//
```

```
1 TEXTINCLUDE
BEGIN
"resource.h"
END
```

```
2 TEXTINCLUDE
BEGIN
#include ""winres.h""\r\n"
#include ""atlsrvres.h""\r\n"
"0"
END
```

```
3 TEXTINCLUDE
BEGIN
```

```
"LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US\r\n"
#pragma code_page(1252)\r\n"
#include ""atlsrv.rc""\r\n"
"0"
```

```
END
```

```
#endif // APSTUDIO_INVOKED
```

```
////////////////////////////////////
//
```

```
// Version
```

```
//
```

```
VS_VERSION_INFO VERSIONINFO
```

```
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGS 0x3FL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
```

```
BLOCK "StringFileInfo"
BEGIN
BLOCK "040904e4"
```

```
BEGIN
VALUE "CompanyName", "TODO: <Company name>"
```

```
VALUE "FileDescription", "TODO: <File description>"
```

```
VALUE "FileVersion", "1.0.0.1"
```

```
VALUE "InternalName", "isapi.dll"
```

```
VALUE "LegalCopyright", "TODO: (c) <Company name>. All rights reserved."
```

```
VALUE "OriginalFilename", "isapi.dll"
```

```
VALUE "ProductName", "TODO: <Product name>"
```

```
VALUE "ProductVersion", "1.0.0.1"
```

```
VALUE "OLESelfRegister", ""
```

```
END
```

```
END
```

```
BLOCK "VarFileInfo"
```

```
BEGIN
```

```
VALUE "Translation", 0x0409, 1252
```

```
END
```

```
END
```

```
LANGUAGE 9, 1
```

```
#pragma code_page(1252)
```

```
////////////////////////////////////
```

```
//
```

```
// String Table
```

```
//
```

```
STRINGTABLE
```

```
BEGIN
```

```
IDS_PROJNAME "tpccIsapi"
```

```
END
```

```
////////////////////////////////////
```

```
#ifndef APSTUDIO_INVOKED
```

```
////////////////////////////////////
```

```
//
```

```
// Generated from the TEXTINCLUDE 3 resource.
```

```
//
```

```
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
```

```
#pragma code_page(1252)
```

```
include "atlsrv.rc"
```

```
////////////////////////////////////
```

```
#endif // not APSTUDIO_INVOKED
```

## tpccIsapi/tpccIsapi.vcproj

```
<?xml version="1.0" encoding="Windows-1252"?>
```

```
<VisualStudioProject
```

```
ProjectType="Visual C++"
```

```

Version="9.00"
Name="tpccsapi"
ProjectGUID="{619C38BD-A8B2-433B-9D21-9A4A64F5C6BD}"
RootNamespace="tpccsapi"
TargetFrameworkVersion="131072"
>
<Platforms>
  <Platform
    Name="Win32"
  />
</Platforms>
<ToolFiles>
</ToolFiles>
<Configurations>
  <Configuration
    Name="Debug|Win32"
    OutputDirectory="Debug"
    IntermediateDirectory="Debug"
    ConfigurationType="2"
  />
  InheritedPropertySheets="$(VCInstallDir)VCProjectDefaults\UpgradeFromVC71.vsprops"
  UseOfATL="1"
  CharacterSet="2"
  >
  <Tool
    Name="VCPreBuildEventTool"
  />
  <Tool
    Name="VCCustomBuildTool"
  />
  <Tool
    Name="VCXMLDataGeneratorTool"
  />
  <Tool
    Name="VCWebServiceProxyGeneratorTool"
  />
  <Tool
    Name="VCMIDLTool"
    PreprocessorDefinitions="_DEBUG"
    MkTypLibCompatible="false"
  />
  <Tool
    Name="VCCLCompilerTool"
    Optimization="0"
    AdditionalIncludeDirectories=".;&quot;C:\home\tpcc_db2\tpc-
c.ibm\include&quot;"
  />
  PreprocessorDefinitions="WIN32; WINDOWS; USRDLL; ATL_ATTRIBUTES; DB2"
  MinimalRebuild="true"
  BasicRuntimeChecks="3"
  RuntimeLibrary="1"
  StructMemberAlignment="0"
  UsePrecompiledHeader="0"
  WarningLevel="3"
  Detect64BitPortabilityProblems="true"
  DebugInformationFormat="4"
  ForcedIncludeFiles=""
  ForcedUsingFiles=""
  ShowIncludes="false"
  />
  <Tool
    Name="VCManagedResourceCompilerTool"
  />
  <Tool
    Name="VCResourceCompilerTool"
    PreprocessorDefinitions="_DEBUG"
    Culture="1033"
  />
  <Tool
    Name="VCPreLinkEventTool"
  />
  <Tool
    Name="VCLinkerTool"

```

```

ShowProgress="0"
OutputFile="$(OutDir)\tpccsapi.dll"
LinkIncremental="2"
ModuleDefinitionFile=".\tpccsapi.def"
GenerateDebugInformation="true"
SubSystem="2"
RandomizedBaseAddress="1"
DataExecutionPrevention="0"
ImportLibrary="$(OutDir)\tpccsapi.lib"
TargetMachine="1"
  />
  <Tool
    Name="VCALinkTool"
  />
  <Tool
    Name="VCManifestTool"
  />
  <Tool
    Name="VCXDCMakeTool"
  />
  <Tool
    Name="VCBscMakeTool"
  />
  <Tool
    Name="VCFxCopTool"
  />
  <Tool
    Name="VCAAppVerifierTool"
  />
  <Tool
    Name="VCPostBuildEventTool"
  />
  </Configuration>
  <Configuration
    Name="Release|Win32"
    OutputDirectory="Release"
    IntermediateDirectory="Release"
    ConfigurationType="2"
  />
  InheritedPropertySheets="$(VCInstallDir)VCProjectDefaults\UpgradeFromVC71.vsprops"
  UseOfMFC="0"
  UseOfATL="0"
  CharacterSet="2"
  >
  <Tool
    Name="VCPreBuildEventTool"
  />
  <Tool
    Name="VCCustomBuildTool"
  />
  <Tool
    Name="VCXMLDataGeneratorTool"
  />
  <Tool
    Name="VCWebServiceProxyGeneratorTool"
  />
  <Tool
    Name="VCMIDLTool"
    PreprocessorDefinitions="NDEBUG"
    MkTypLibCompatible="false"
  />
  <Tool
    Name="VCCLCompilerTool"
    Optimization="2"
    InlineFunctionExpansion="0"
    EnableIntrinsicFunctions="false"
    FavorSizeOrSpeed="1"
    OmitFramePointers="true"
    EnableFiberSafeOptimizations="false"
    AdditionalIncludeDirectories=".;\tpccCommon;.\tpc-c.ibm\include"
    AdditionalUsingDirectories=""
  />
  PreprocessorDefinitions="WIN32; WINDOWS; NDEBUG; USRDLL; ATL_ATTRIBUTES; DB2"

```

```

RuntimeLibrary="0"
UsePrecompiledHeader="0"
WarningLevel="3"
Detect64BitPortabilityProblems="true"
DebugInformationFormat="3"
  />
  <Tool
    Name="VCManagedResourceCompilerTool"
  />
  <Tool
    Name="VCResourceCompilerTool"
    PreprocessorDefinitions="NDEBUG"
    Culture="1033"
  />
  <Tool
    Name="VCPreLinkEventTool"
  />
  <Tool
    Name="VCLinkerTool"
    OutputFile="$(OutDir)\tpccsapi.dll"
    LinkIncremental="1"
    AdditionalLibraryDirectories="&quot;C:\Program Files (x86)\Microsoft
Visual Studio 9.0\VC\lib&quot;;&quot;C:\Program Files\Microsoft
SDKs\Windows\v6.0A\lib&quot;;C:\windows\system32"
    IgnoreDefaultLibraryNames=""
    ModuleDefinitionFile=".\tpccsapi.def"
    GenerateDebugInformation="true"
    SubSystem="2"
    StackReserveSize="256000"
    StackCommitSize="256000"
    OptimizeReferences="2"
    EnableCOMDATFolding="2"
    RandomizedBaseAddress="1"
    DataExecutionPrevention="0"
    ImportLibrary="$(OutDir)\tpccsapi.lib"
    TargetMachine="1"
  />
  <Tool
    Name="VCALinkTool"
  />
  <Tool
    Name="VCManifestTool"
    EmbedManifest="true"
  />
  <Tool
    Name="VCXDCMakeTool"
  />
  <Tool
    Name="VCBscMakeTool"
  />
  <Tool
    Name="VCFxCopTool"
  />
  <Tool
    Name="VCAAppVerifierTool"
  />
  <Tool
    Name="VCPostBuildEventTool"
  />
  </Configuration>
  </Configurations>
  <References>
  </References>
  <Files>
    <Filter
      Name="Source Files"
      Filter="*.cpp;*.c;*.cxx;*.def;*.idl;*.h;*.hpp;*.bat;*.asm;*.asmx"
      UniqueIdentifier="{4FC737F1-C7A5-4376-A066-2A32D752A2FF}"
    >
    <File
      RelativePath=".\htmlPhraser.cpp"
    >
  </File>

```



```

<File
  RelativePath=".StdAfx.cpp"
>
<FileConfiguration
  Name="Debug|Win32"
>
  <Tool
    Name="VCCLCompilerTool"
    UsePrecompiledHeader="1"
  />
</FileConfiguration>
<FileConfiguration
  Name="Release|Win32"
>
  <Tool
    Name="VCCLCompilerTool"
    UsePrecompiledHeader="1"
  />
</FileConfiguration>
</File>
<File
  RelativePath=".time.cpp"
>
</File>
<File
  RelativePath=".tpccsapi.cpp"
>
</File>
<File
  RelativePath=".tpccsapi.def"
>
</File>
</Filter>
<Filter
  Name="Header Files"
  Filter="*.h;*.hpp;*.hxx;*.hmi;*.inc;*.xsd"
  UniqueIdentifier="{93995380-89BD-4b04-88EB-625FBE52EBFB}"
>
  <File
    RelativePath=".htmlPhraser.h"
  >
</File>
<File
  RelativePath=".resource.h"
>
</File>
<File
  RelativePath=".StdAfx.h"
>
</File>
<File
  RelativePath=".tpcc.h"
>
</File>
<File
  RelativePath=".tpccsapi.hpp"
>
</File>
</Filter>
<Filter
  Name="Resource Files"
  Filter="*.rc;*.ico;*.cur;*.bmp;*.dlg;*.rc2;*.rct;*.bin;*.rgs;*.gif;*.jpg;*.jpeg;*.jpe;*.res;*.x"
  UniqueIdentifier="{67DA6AB6-F800-4c08-8B7A-83BB121AAD01}"
>
  <File
    RelativePath=".tpccsapi.rc"
  >
</File>
</Filter>
<File
  RelativePath=".ReadMe.txt"
>
</File>

```

```

</Files>
</Globals>
</Globals>
</VisualStudioProject>

```

## A.2 Client Transaction Code

### Makefile.config

```

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

#
# Makefile.config - AIX 64-bit
#
#
# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g", "-DDEBUGIT" "-g -DDEBUGIT" or left blank
CC=xlc
CFLAGS_OS=-qflag=i-i -qlanglvl=ansi -qplcsomt -DSQLUNIX -DSQLAIX -q64 -O3 -D_LARGE_FILES
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=xlc
LD_STORP=xlc
LD_FLAGS_EXEC=-lm -q64
LD_FLAGS_SHLIB=qmkshrobj
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB) -bE.$@.exp -lc -b64
LD_FLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2
LD_FLAGS_OUT=-o

# Library Configuration
AR=ar
ARFLAGS=-r -v -X64
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=.a
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
# #####

PRP_OPTS = PACKAGE \
           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

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

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

```

```
.sqc.c:
@echo "Prepping $.sqc"
db2 prep $.sqc $(PRP_OPTS)
db2 grant execute on package TPCCCTX to public

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

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

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

### Src.Common/tpccctx.sqc

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

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

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

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

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

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

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

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

```
EXEC SQL CONNECT TO :dbname IN SHARE MODE USER :username USING :password;
}

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

return ConnectSQLCODE;
}

return 0;
}

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

EXEC SQL CONNECT RESET;

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

if (DisconnectSQLCODE) {
return DisconnectSQLCODE;
}
}

return 0;
}
```

### Src.Common/tpccdbg.c

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

```
/*
* tpccdbg.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 {
strncpy(debugPath, "/tmp");
}
strcat(debugPath, "/7");
}
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_fn, "-----\n");
fprintf(err_fn, "Transaction: %s (%s)\n", tranName, msg);
fprintf(err_fn, "FILE %s (%u)\n", file, line);
fprintf(err_fn, "SQLCODE %d", psqlca->sqlcode);
fprintf(err_fn, "PID %d", getpid());
fprintf(err_fn, "TIME %s\n", timeStamp);
fprintf(err_fn, "-----\n");
fprintf(err_fn, "%s", errStr);
fprintf(err_fn, "-----\n");

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

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

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

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

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

fprintf(err_fn, "\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, "PID %d", getpid());
    fprintf(debug_fp, "\n=====");

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

    fprintf(debug_fp, "out_delivery_struct {\n");
    fprintf(debug_fp, "its_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, "its_O_ID[%d] = %d\n",
                j, delivery_ptr->s_O_ID[j]);
    }
    fprintf(debug_fp, "t}\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, "PID %d", getpid());
    fprintf(debug_fp, "\n=====");

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

    fprintf(debug_fp, "its_C_ID = %d (%X)\n",
            in_neword->s_C_ID, in_neword->s_C_ID);
    fprintf(debug_fp, "its_W_ID = %d (%X)\n",
            in_neword->s_W_ID, in_neword->s_W_ID);
    fprintf(debug_fp, "its_D_ID = %d (%X)\n",
            in_neword->s_D_ID, in_neword->s_D_ID);
    fprintf(debug_fp, "its_O_OL_CNT = %d (%X)\n",
            in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
    fprintf(debug_fp, "its_all_local = %d (%X)\n",
            in_neword->s_all_local, in_neword->s_all_local);
    // fprintf(debug_fp, "its_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, "tits_OL_ID[%d] = %d (%X)\n",
                j, in_neword->in_item[j].s_OL_ID, in_neword->in_item[j].s_OL_ID);
        fprintf(debug_fp, "tits_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, "tits_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");

    fprintf(debug_fp, "out_neword_struct {\n");
    fprintf(debug_fp, "its_C_LAST = %s\n",
            neword_ptr->s_C_LAST);
    fprintf(debug_fp, "its_C_CREDIT = %s\n",
            neword_ptr->s_C_CREDIT);
    fprintf(debug_fp, "its_W_TAX = %04.4f\n",
            neword_ptr->s_W_TAX);
    fprintf(debug_fp, "its_D_TAX = %04.4f\n",
            neword_ptr->s_D_TAX);
    fprintf(debug_fp, "its_C_DISCOUNT = %04.4f\n",

```

```

neword_ptr->s_C_DISCOUNT);
fprintf(debug_fp,"ts_O_ID = %d (%X)\n",
neword_ptr->s_O_ID, neword_ptr->s_O_ID);
fprintf(debug_fp,"ts_O_OL_CNT = %d (%X)\n",
neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
fprintf(debug_fp,"ts_O_ENTRY_D = %s\n",
neword_ptr->s_O_ENTRY_D_time);
fprintf(debug_fp,"ts_total_amount = %2f\n",
neword_ptr->s_total_amount);
fprintf(debug_fp,"ts_transtatus = %d (%X)\n",
neword_ptr->s_transtatus, neword_ptr->s_transtatus);
fprintf(debug_fp,"tdeadlocks = %d (%X)\n",
neword_ptr->deadlocks, neword_ptr->deadlocks);

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

fprintf(debug_fp,"titems {\n");
items = neword_ptr->s_O_OL_CNT;
for (j=0; j<items; j++) {
    if(j != 0)
        fprintf(debug_fp,"n");
    fprintf(debug_fp,"tts_I_NAME[%d] = %s\n",
j, neword_ptr->item[j].s_I_NAME);
    fprintf(debug_fp,"tts_I_PRICE[%d] = %2f\n",
j, neword_ptr->item[j].s_I_PRICE);
    fprintf(debug_fp,"tts_OL_AMOUNT[%d] = %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,"tj)\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=====");

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, "j)\n");

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

fprintf(debug_fp, "titems {\n");
items = ordstat_ptr->s_ol_cnt;
for (j = 0; j < items; j++) {
    if(j != 0)
        fprintf(debug_fp, "n");
    fprintf(debug_fp, "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");
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=====");

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, "j)\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 = %04f\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");

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,time(&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 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****/

/*
 *
 * tpcmisc.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
#####

PRP_OPTS = PACKAGE \
           ISOLATION RR \
           QUERYOPT 7 \
           EXPLAIN ALL \
           MESSAGES $*.prep.msg

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

CFLAGS = $(CFLAGS_OS) $(INCLUDE) $(CFLAGS_DEBUG) \
         -D$(DB2EDITION) -D$(DB2VERSION) \
         -D$SQLA_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 -tvf cat-proc.ddl +o -z cat-proc.out
-db2 -td% -vf cat-func.ddl +o -z cat-func.out

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

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

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

connect:
-db2 connect to $(TPCC_DBNAME)

disconnect:
-db2 connect reset
-db2 terminate

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

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

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

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

.SUFFIXES: $(OBJEXT) .c .sqc

tpcc_all_sql.c:
@echo "Prepping $*.sqc"
db2 prep $*.sqc $(PRP_OPTS)
db2 grant execute on package TPCC_ALL to public

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
) 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_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
)
;

/* Determine the C_ID */
SET ( C_ID
    = ( SELECT C_ID
        FROM ( SELECT C_ID
              , COUNT(*) OVER() AS COUNT
              , ROWNUMBER() OVER (ORDER BY C_FIRST) AS NUM
            FROM CUSTOMER
            WHERE C_LAST = PAY_C_LAST.C_LAST
            AND C_W_ID = PAY_C_LAST.C_W_ID
            AND C_D_ID = PAY_C_LAST.C_D_ID
          ) AS T
        WHERE NUM = (COUNT + BIGINT(1)) / BIGINT(2)
    )
;

/* Update the middle customer */
SET ( C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
    , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
    , C_DISCOUNT, C_BALANCE, C_DATA )
    = ( SELECT C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
        , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
        , C_DISCOUNT, C_BALANCE
        , CASE WHEN C_CREDIT = 'BC' THEN SUBSTR(C_DATA, 1, 200) ELSE NULL END AS
        C_DATA
      FROM NEW TABLE ( UPDATE CUSTOMER
        SET C_BALANCE = C_BALANCE - PAY_C_LAST.H_AMOUNT
        , C_YTD_PAYMENT = C_YTD_PAYMENT + PAY_C_LAST.H_AMOUNT
        , C_PAYMENT_CNT = C_PAYMENT_CNT + SMALLINT(1)
        , C_DATA = CASE WHEN C_CREDIT = 'BC'
            THEN CHAR(C_ID) -- 11 bytes long
            || BAD_CREDIT_PREFIX -- 28 bytes long
            || SUBSTR(C_DATA, 1, 461) -- 461 + 39 = 500
            ELSE C_DATA
          END
        WHERE C_W_ID = PAY_C_LAST.C_W_ID
        AND C_D_ID = PAY_C_LAST.C_D_ID
        AND C_ID = VAR_C_ID
      ) AS U
)
;

/* Update the warehouse */
SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE, W_ZIP
    = ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE, W_ZIP
      FROM OLD TABLE ( UPDATE WAREHOUSE
        SET W_YTD = W_YTD + PAY_C_LAST.H_AMOUNT
        WHERE W_ID = PAY_C_LAST.W_ID
      ) AS U
    )
;

```

```

/* Finally insert into the warehouse */
INSERT
INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID, H_W_ID, H_DATA, H_DATE, H_AMOUNT
)
VALUES ( VAR_C_ID
    , PAY_C_LAST.C_D_ID
    , PAY_C_LAST.C_W_ID
    , PAY_C_LAST.D_ID
    , PAY_C_LAST.W_ID
    , VAR.W_NAME || CHAR(' ', 4) || VAR.D_NAME
    , VAR.H_DATE
    , PAY_C_LAST.H_AMOUNT
)
;

/* Done - return the collected data */
RETURN VALUES ( W_STREET_1, W_STREET_2, W_CITY, W_STATE, W_ZIP
    , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
    , C_ID, C_FIRST, C_MIDDLE, C_STREET_1, C_STREET_2
    , C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT, C_CREDIT_LIM
    , C_DISCOUNT, C_BALANCE, C_DATA, H_DATE
)
;

END
%

CREATE FUNCTION PAY_C_ID( W_ID INTEGER
    , D_ID SMALLINT
    , C_W_ID INTEGER
    , C_D_ID SMALLINT
    , C_ID INTEGER
    , H_AMOUNT DECIMAL(6,2)
    , BAD_CREDIT_PREFIX VARCHAR(34)
)
RETURNS TABLE( W_STREET_1 CHAR(20)
    , W_STREET_2 CHAR(20)
    , W_CITY CHAR(20)
    , W_STATE CHAR(2)
    , W_ZIP CHAR(9)
    , D_STREET_1 CHAR(20)
    , D_STREET_2 CHAR(20)
    , D_CITY CHAR(20)
    , D_STATE CHAR(2)
    , D_ZIP CHAR(9)
    , C_LAST VARCHAR(16)
    , C_FIRST VARCHAR(16)
    , C_MIDDLE CHAR(2)
    , C_STREET_1 VARCHAR(20)
    , C_STREET_2 VARCHAR(20)
    , C_CITY VARCHAR(20)
    , C_STATE CHAR(2)
    , C_ZIP CHAR(9)
    , C_PHONE CHAR(16)
    , C_SINCE TIMESTAMP
    , C_CREDIT CHAR(2)
    , C_CREDIT_LIM DECIMAL(12,2)
    , C_DISCOUNT REAL
    , C_BALANCE DECIMAL(12,2)
    , C_DATA CHAR(200)
    , H_DATE TIMESTAMP
)
SPECIFIC PAY_C_ID
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL
VAR: BEGIN ATOMIC

```

```

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

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

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

DECLARE C_LAST VARCHAR(16);

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

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

/* Update District and 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
, SUPP_W_ID INT
, O_ID INT
, D_ID SMALLINT
)
RETURNS TABLE( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, OL_DIST_INFO CHAR(24)
, S_DATA VARCHAR(50)
)

```

```

, S_QUANTITY SMALLINT
)
SPECIFIC NEW_OL_ALL
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE I_PRICE DECIMAL(5,2);
DECLARE I_NAME CHAR(24);
DECLARE I_DATA VARCHAR(50);
DECLARE OL_DIST_INFO CHAR(24);
DECLARE S_DATA VARCHAR(50);
DECLARE S_QUANTITY SMALLINT;

SET ( I_PRICE, I_NAME, I_DATA )
= ( SELECT
I_PRICE
, I_NAME
, I_DATA
FROM ITEM
WHERE ITEM.I_ID = NEW_OL_ALL.I_ID
);

SET ( OL_DIST_INFO, S_DATA, S_QUANTITY )
= ( SELECT OL_DIST_INFO
, S_DATA
, S_QUANTITY
FROM NEW TABLE ( UPDATE STOCK
INCLUDE ( OL_DIST_INFO CHAR(24) )
SET S_QUANTITY = CASE WHEN S_QUANTITY - NEW_OL_ALL.I_QTY >= 10
THEN S_QUANTITY - NEW_OL_ALL.I_QTY
ELSE S_QUANTITY - NEW_OL_ALL.I_QTY + 91
END
, S_ORDER_CNT = S_ORDER_CNT + SMALLINT( 1 )
, S_YTD = S_YTD + NEW_OL_ALL.I_QTY
, S_REMOTE_CNT = CASE WHEN NEW_OL_ALL.SUPP_W_ID =
NEW_OL_ALL.W_ID
THEN S_REMOTE_CNT
ELSE S_REMOTE_CNT + SMALLINT( 1 )
END
, OL_DIST_INFO = CASE D_ID WHEN SMALLINT( 1 ) THEN
WHEN SMALLINT( 2 ) THEN S_DIST_02
WHEN SMALLINT( 3 ) THEN S_DIST_03
WHEN SMALLINT( 4 ) THEN S_DIST_04
WHEN SMALLINT( 5 ) THEN S_DIST_05
WHEN SMALLINT( 6 ) THEN S_DIST_06
WHEN SMALLINT( 7 ) THEN S_DIST_07
WHEN SMALLINT( 8 ) THEN S_DIST_08
WHEN SMALLINT( 9 ) THEN S_DIST_09
WHEN SMALLINT( 10 ) THEN S_DIST_10
END
WHERE S_I_ID = NEW_OL_ALL.I_ID
AND S_W_ID = NEW_OL_ALL.SUPP_W_ID
) AS U
);

RETURN VALUES( VAR.I_PRICE

```

```

, VAR.I_NAME
, VAR.I_DATA
, VAR.OL_DIST_INFO
, VAR.S_DATA
, VAR.S_QUANTITY
)
;

END
%

CREATE FUNCTION NEW_OL_LOCAL( I_ID INT
, I_QTY SMALLINT
, W_ID INT
, O_ID INT
, D_ID SMALLINT
)
RETURNS TABLE( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, OL_DIST_INFO CHAR(24)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT
)
SPECIFIC NEW_OL_LOCAL
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE I_PRICE DECIMAL(5,2);
DECLARE I_NAME CHAR(24);
DECLARE I_DATA VARCHAR(50);
DECLARE OL_DIST_INFO CHAR(24);
DECLARE S_DATA VARCHAR(50);
DECLARE S_QUANTITY SMALLINT;

SET ( I_PRICE, I_NAME, I_DATA )
= ( SELECT
I_PRICE
, I_NAME
, I_DATA
FROM ITEM
WHERE ITEM.I_ID = NEW_OL_LOCAL.I_ID
);

SET ( OL_DIST_INFO, S_DATA, S_QUANTITY )
= ( SELECT OL_DIST_INFO
, S_DATA
, S_QUANTITY
FROM NEW TABLE ( UPDATE STOCK
INCLUDE ( OL_DIST_INFO CHAR(24) )
SET S_QUANTITY = CASE WHEN S_QUANTITY - NEW_OL_LOCAL.I_QTY >= 10
THEN S_QUANTITY - NEW_OL_LOCAL.I_QTY
ELSE S_QUANTITY - NEW_OL_LOCAL.I_QTY + 91
END
, S_ORDER_CNT = S_ORDER_CNT + SMALLINT( 1 )
, S_YTD = S_YTD + NEW_OL_LOCAL.I_QTY
, OL_DIST_INFO = CASE D_ID WHEN SMALLINT( 1 ) THEN
S_DIST_01

```

```

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

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

END
%

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

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

SPECIFIC NEW_WH

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

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

SET O_ENTRY_D = CURRENT_TIMESTAMP;

INSERT

INTO NEW_ORDER ( NO_O_ID, NO_D_ID, NO_W_ID )

VALUES ( O_ID
, D_ID
, W_ID
)
;

INSERT

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

```

```

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

SET ( C_DISCOUNT, C_LAST, C_CREDIT )
= ( SELECT C_DISCOUNT, C_LAST, C_CREDIT

FROM CUSTOMER

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

SET W_TAX
= ( SELECT W_TAX

FROM WAREHOUSE

WHERE W_ID = NEW_WH.W_ID
)
;

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

```

```

END
%

```

### Src.Srv/cat-proc.ddl

```

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

```

### Src.Srv/tpcc\_all\_sql.sqc

```

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

```

```

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

```

```

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

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

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

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

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

SQL_API_RC new_order_internal( char *pin, char *pout )
{
    struct out_neword_struct *neword;

    struct in_neword_struct *in_neword;

    struct sqlca sqlca;

    int  fbadItemDetected = 0;

    EXEC SQL BEGIN DECLARE SECTION;

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

    sqlint32 w_id;
    short  d_id;
    sqlint32 c_id;

    sqlint32 next_o_id;

    short  s_quantity;

    sqlint32 supply_w_id;

    short  inputItemCount;

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

    char  o_entry_d [27];

    short  allLocal;

    float  item_price;

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

```

```

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

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

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

EXEC SQL END DECLARE SECTION;

int storedProcRc;
int inputItemArrayIndex;

char stockDistrictInformationArray [15][25];

#define stockDistrictInformation stockDistrictInformationArray[ inputItemArrayIndex ]

// Redirected input fields

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

#define inputItemCount in_neword->s_OL_OL_CNT

#define allLocal in_neword->s_all_local

// Redirected output fields

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

// This output field becomes an input field to order_line

#define next_o_id neword->s_O_ID

// item price/name

#define item_name neword->item[ inputItemArrayIndex ]s_I_NAME

float i_priceArray[ 15 ];

#define item_price i_priceArray[ inputItemArrayIndex ]

// Handle the generic/brand distinction

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

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

// Redirect hostvars to input structure

#define id0 in_neword->in_item[0]s_OL_I_ID
#define id1 in_neword->in_item[1]s_OL_I_ID
#define id2 in_neword->in_item[2]s_OL_I_ID
#define id3 in_neword->in_item[3]s_OL_I_ID
#define id4 in_neword->in_item[4]s_OL_I_ID
#define id5 in_neword->in_item[5]s_OL_I_ID
#define id6 in_neword->in_item[6]s_OL_I_ID
#define id7 in_neword->in_item[7]s_OL_I_ID

```

```

#define id8 in_neword->in_item[8]s_OL_I_ID
#define id9 in_neword->in_item[9]s_OL_I_ID
#define id10 in_neword->in_item[10]s_OL_I_ID
#define id11 in_neword->in_item[11]s_OL_I_ID
#define id12 in_neword->in_item[12]s_OL_I_ID
#define id13 in_neword->in_item[13]s_OL_I_ID
#define id14 in_neword->in_item[14]s_OL_I_ID

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

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

EXEC SQL DECLARE ISOL_Remote_1 CURSOR FOR

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

FROM Table( VALUES

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

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

, TABLE( NEW_OL_ALL( L_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID

```

```

, O_ID
, D_ID
)
) AS NEW_OL_ALL

WHERE NEW_OL_ALL.L_PRICE IS NOT NULL

)

SELECT L_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
, L_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, L_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
, L_ID
, I_SUPPLY_W_ID
, (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
, I_QTY
, (L_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, L_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
, L_ID
, I_SUPPLY_W_ID
, I_QTY

FROM Table( VALUES

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

```

```

        ,( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
    ) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_3 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID

```

```

, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
, ( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
, ( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
)
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_4 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER

```

```

, I_ID
, I_SUPPLY_W_ID
, (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
, ( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
, ( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
, ( SMALLINT(4) ,:id3 ,:ol_quantity3 ,:supply_w_id3 )
)
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

```

```

FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_5 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,(TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,w_id AS W_ID
,d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3 ,:supply_w_id3 )
,( SMALLINT(5) ,:id4 ,:ol_quantity4 ,:supply_w_id4 )
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
,I_QTY
,W_ID
,I_SUPPLY_W_ID
,O_ID
,D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE DECIMAL(5,2)
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT )

```

```

SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_6 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,(TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,w_id AS W_ID
,d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3 ,:supply_w_id3 )
,( SMALLINT(5) ,:id4 ,:ol_quantity4 ,:supply_w_id4 )
,( SMALLINT(6) ,:id5 ,:ol_quantity5 ,:supply_w_id5 )
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
,I_QTY
,W_ID
,I_SUPPLY_W_ID
,O_ID
,D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID

```

```

,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE DECIMAL(5,2)
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT )
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_7 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,(TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,w_id AS W_ID
,d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table( VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3 ,:supply_w_id3 )
,( SMALLINT(5) ,:id4 ,:ol_quantity4 ,:supply_w_id4 )
,( SMALLINT(6) ,:id5 ,:ol_quantity5 ,:supply_w_id5 )
,( SMALLINT(7) ,:id6 ,:ol_quantity6 ,:supply_w_id6 )
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
,I_QTY
,W_ID
,I_SUPPLY_W_ID
,O_ID
,D_ID
)
) AS NEW_OL_ALL

```

```

WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)

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

FROM NEW TABLE ( INSERT INTO ORDER_LINE

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

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

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

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Remote_8 CURSOR FOR

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

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

FROM Table( VALUES

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

) 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 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 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 DATA

) AS INS
;

```

```

( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
, ( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
, ( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )

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

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

WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

)

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

FROM NEW TABLE ( INSERT INTO ORDER_LINE

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

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

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

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Remote_9 CURSOR FOR

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

FROM 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 DATA

) AS INS
;

```

```

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_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 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 DATA

) AS INS
;

```

```

;
EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,(TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,w_id AS W_ID
,d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table(VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3 ,:supply_w_id3 )
,( SMALLINT(5) ,:id4 ,:ol_quantity4 ,:supply_w_id4 )
,( SMALLINT(6) ,:id5 ,:ol_quantity5 ,:supply_w_id5 )
,( SMALLINT(7) ,:id6 ,:ol_quantity6 ,:supply_w_id6 )
,( SMALLINT(8) ,:id7 ,:ol_quantity7 ,:supply_w_id7 )
,( SMALLINT(9) ,:id8 ,:ol_quantity8 ,:supply_w_id8 )
,( SMALLINT(10) ,:id9 ,:ol_quantity9 ,:supply_w_id9 )
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
,I_QTY
,W_ID
,I_SUPPLY_W_ID
,O_ID
,D_ID
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE, I_NAME, I_DATA, OL_DIST_INFO, S_DATA, S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE DECIMAL(5,2)
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)

```

```

,S_QUANTITY SMALLINT)
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_11 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,(TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,w_id AS W_ID
,d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table(VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3 ,:supply_w_id3 )
,( SMALLINT(5) ,:id4 ,:ol_quantity4 ,:supply_w_id4 )
,( SMALLINT(6) ,:id5 ,:ol_quantity5 ,:supply_w_id5 )
,( SMALLINT(7) ,:id6 ,:ol_quantity6 ,:supply_w_id6 )
,( SMALLINT(8) ,:id7 ,:ol_quantity7 ,:supply_w_id7 )
,( SMALLINT(9) ,:id8 ,:ol_quantity8 ,:supply_w_id8 )
,( SMALLINT(10) ,:id9 ,:ol_quantity9 ,:supply_w_id9 )
,( SMALLINT(11) ,:id10 ,:ol_quantity10 ,:supply_w_id10 )
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE(NEW_OL_ALL( I_ID
,I_QTY
,W_ID
,I_SUPPLY_W_ID
,O_ID
,D_ID
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)
SELECT I_PRICE, I_NAME, I_DATA, OL_DIST_INFO, S_DATA, S_QUANTITY

```

```

FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
,OL_D_ID
,OL_W_ID
,OL_NUMBER
,OL_I_ID
,OL_SUPPLY_W_ID
,OL_DELIVERY_D
,OL_QUANTITY
,OL_AMOUNT
,OL_DIST_INFO
)
INCLUDE( I_PRICE DECIMAL(5,2)
,I_NAME CHAR(24)
,I_DATA VARCHAR(50)
,S_DATA VARCHAR(50)
,S_QUANTITY SMALLINT)
SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,OL_DELIVERY_D
,I_QTY
,TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_12 CURSOR FOR
WITH DATA AS ( SELECT O_ID
,D_ID
,W_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,(TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
,I_QTY
,(I_PRICE * I_QTY) AS TOTAL_PRICE
,OL_DIST_INFO
,I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
,w_id AS W_ID
,d_id as D_ID
,OL_NUMBER
,I_ID
,I_SUPPLY_W_ID
,I_QTY
FROM Table(VALUES
( SMALLINT(1) ,:id0 ,:ol_quantity0 ,:supply_w_id0 )
,( SMALLINT(2) ,:id1 ,:ol_quantity1 ,:supply_w_id1 )
,( SMALLINT(3) ,:id2 ,:ol_quantity2 ,:supply_w_id2 )
,( SMALLINT(4) ,:id3 ,:ol_quantity3 ,:supply_w_id3 )
,( SMALLINT(5) ,:id4 ,:ol_quantity4 ,:supply_w_id4 )
,( SMALLINT(6) ,:id5 ,:ol_quantity5 ,:supply_w_id5 )
,( SMALLINT(7) ,:id6 ,:ol_quantity6 ,:supply_w_id6 )
,( SMALLINT(8) ,:id7 ,:ol_quantity7 ,:supply_w_id7 )
,( SMALLINT(9) ,:id8 ,:ol_quantity8 ,:supply_w_id8 )
,( SMALLINT(10) ,:id9 ,:ol_quantity9 ,:supply_w_id9 )
,( SMALLINT(11) ,:id10 ,:ol_quantity10 ,:supply_w_id10 )
,( SMALLINT(12) ,:id11 ,:ol_quantity11 ,:supply_w_id11 )

```

```

) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
) )
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_13 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID

```

```

, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 , :supply_w_id0 )
, ( SMALLINT( 2) , :id1 , :ol_quantity1 , :supply_w_id1 )
, ( SMALLINT( 3) , :id2 , :ol_quantity2 , :supply_w_id2 )
, ( SMALLINT( 4) , :id3 , :ol_quantity3 , :supply_w_id3 )
, ( SMALLINT( 5) , :id4 , :ol_quantity4 , :supply_w_id4 )
, ( SMALLINT( 6) , :id5 , :ol_quantity5 , :supply_w_id5 )
, ( SMALLINT( 7) , :id6 , :ol_quantity6 , :supply_w_id6 )
, ( SMALLINT( 8) , :id7 , :ol_quantity7 , :supply_w_id7 )
, ( SMALLINT( 9) , :id8 , :ol_quantity8 , :supply_w_id8 )
, ( SMALLINT( 10) , :id9 , :ol_quantity9 , :supply_w_id9 )
, ( SMALLINT( 11) , :id10 , :ol_quantity10 , :supply_w_id10 )
, ( SMALLINT( 12) , :id11 , :ol_quantity11 , :supply_w_id11 )
, ( SMALLINT( 13) , :id12 , :ol_quantity12 , :supply_w_id12 )
) AS X ( OL_NUMBER , I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
) )
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT NULL
)

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_14 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, (TIMESTAMP('0001-01-01 00:00:00')) AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID

```

```

) 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
,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 )

) 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
,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 )

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

```

```

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
) 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
) 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 DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_9 CURSOR FOR

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

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

    FROM Table(VALUES

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

```

```

      , ( SMALLINT(2) , :id1 , :ol_quantity1 )
    , ( SMALLINT(3) , :id2 , :ol_quantity2 )
    , ( SMALLINT(4) , :id3 , :ol_quantity3 )
    , ( SMALLINT(5) , :id4 , :ol_quantity4 )
    , ( SMALLINT(6) , :id5 , :ol_quantity5 )
    , ( SMALLINT(7) , :id6 , :ol_quantity6 )
    , ( SMALLINT(8) , :id7 , :ol_quantity7 )
    , ( SMALLINT(9) , :id8 , :ol_quantity8 )

    ) AS X ( OL_NUMBER, I_ID , I_QTY
  ) AS ITEMLIST
  , TABLE(NEW_OL_LOCAL( I_ID
    , I_QTY
    , W_ID
    , O_ID
    , D_ID
  ) AS NEW_OL_LOCAL

  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

)

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

FROM NEW TABLE ( INSERT INTO ORDER_LINE

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

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

  SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , 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 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

```

```

, (L_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
, (L_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

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

FROM Table( VALUES

( SMALLINT( 1) ,:id0 ,:ol_quantity0 )
, ( SMALLINT( 2) ,:id1 ,:ol_quantity1 )
, ( SMALLINT( 3) ,:id2 ,:ol_quantity2 )
, ( SMALLINT( 4) ,:id3 ,:ol_quantity3 )
, ( SMALLINT( 5) ,:id4 ,:ol_quantity4 )
, ( SMALLINT( 6) ,:id5 ,:ol_quantity5 )
, ( SMALLINT( 7) ,:id6 ,:ol_quantity6 )
, ( SMALLINT( 8) ,:id7 ,:ol_quantity7 )
, ( SMALLINT( 9) ,:id8 ,:ol_quantity8 )
, ( SMALLINT( 10) ,:id9 ,:ol_quantity9 )
, ( SMALLINT( 11) ,:id10 ,:ol_quantity10 )

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

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)

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

FROM NEW TABLE ( INSERT INTO ORDER_LINE

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

INCLUDE( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)

```

```

, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)

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

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_12 CURSOR FOR

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

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

FROM Table( VALUES

( SMALLINT( 1) ,:id0 ,:ol_quantity0 )
, ( SMALLINT( 2) ,:id1 ,:ol_quantity1 )
, ( SMALLINT( 3) ,:id2 ,:ol_quantity2 )
, ( SMALLINT( 4) ,:id3 ,:ol_quantity3 )
, ( SMALLINT( 5) ,:id4 ,:ol_quantity4 )
, ( SMALLINT( 6) ,:id5 ,:ol_quantity5 )
, ( SMALLINT( 7) ,:id6 ,:ol_quantity6 )
, ( SMALLINT( 8) ,:id7 ,:ol_quantity7 )
, ( SMALLINT( 9) ,:id8 ,:ol_quantity8 )
, ( SMALLINT( 10) ,:id9 ,:ol_quantity9 )
, ( SMALLINT( 11) ,:id10 ,:ol_quantity10 )
, ( SMALLINT( 12) ,:id11 ,:ol_quantity11 )

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

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL
)

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

```

```

FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE( I_PRICE DECIMAL(5,2)
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)

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

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_13 CURSOR FOR

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

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

FROM Table( VALUES

( SMALLINT( 1 ) ,:id0 ,:ol_quantity0 )
, ( SMALLINT( 2 ) ,:id1 ,:ol_quantity1 )
, ( SMALLINT( 3 ) ,:id2 ,:ol_quantity2 )
, ( SMALLINT( 4 ) ,:id3 ,:ol_quantity3 )
, ( SMALLINT( 5 ) ,:id4 ,:ol_quantity4 )
, ( SMALLINT( 6 ) ,:id5 ,:ol_quantity5 )
, ( SMALLINT( 7 ) ,:id6 ,:ol_quantity6 )
, ( SMALLINT( 8 ) ,:id7 ,:ol_quantity7 )
, ( SMALLINT( 9 ) ,:id8 ,:ol_quantity8 )
, ( SMALLINT( 10 ) ,:id9 ,:ol_quantity9 )
, ( SMALLINT( 11 ) ,:id10 ,:ol_quantity10 )
, ( SMALLINT( 12 ) ,:id11 ,:ol_quantity11 )

```

```

, ( SMALLINT( 13 ) ,:id12 ,:ol_quantity12 )
) AS X ( OL_NUMBER , I_ID , I_QTY )
) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

)

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

FROM NEW TABLE ( INSERT INTO ORDER_LINE

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

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

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

FROM DATA

) AS INS
;

EXEC SQL DECLARE ISOL_Local_14 CURSOR FOR

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

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

```

```

, OL_NUMBER
, I_ID
, I_QTY

FROM Table( VALUES

( SMALLINT( 1 ) ,:id0 ,:ol_quantity0 )
, ( SMALLINT( 2 ) ,:id1 ,:ol_quantity1 )
, ( SMALLINT( 3 ) ,:id2 ,:ol_quantity2 )
, ( SMALLINT( 4 ) ,:id3 ,:ol_quantity3 )
, ( SMALLINT( 5 ) ,:id4 ,:ol_quantity4 )
, ( SMALLINT( 6 ) ,:id5 ,:ol_quantity5 )
, ( SMALLINT( 7 ) ,:id6 ,:ol_quantity6 )
, ( SMALLINT( 8 ) ,:id7 ,:ol_quantity7 )
, ( SMALLINT( 9 ) ,:id8 ,:ol_quantity8 )
, ( SMALLINT( 10 ) ,:id9 ,:ol_quantity9 )
, ( SMALLINT( 11 ) ,:id10 ,:ol_quantity10 )
, ( SMALLINT( 12 ) ,:id11 ,:ol_quantity11 )
, ( SMALLINT( 13 ) ,:id12 ,:ol_quantity12 )
, ( SMALLINT( 14 ) ,:id13 ,:ol_quantity13 )

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

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

)

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

FROM NEW TABLE ( INSERT INTO ORDER_LINE

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

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

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

FROM DATA

) AS INS

```

```

;
EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

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

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

FROM Table VALUES

( SMALLINT( 1 ), :id0 , :ol_quantity0 )
, ( SMALLINT( 2 ), :id1 , :ol_quantity1 )
, ( SMALLINT( 3 ), :id2 , :ol_quantity2 )
, ( SMALLINT( 4 ), :id3 , :ol_quantity3 )
, ( SMALLINT( 5 ), :id4 , :ol_quantity4 )
, ( SMALLINT( 6 ), :id5 , :ol_quantity5 )
, ( SMALLINT( 7 ), :id6 , :ol_quantity6 )
, ( SMALLINT( 8 ), :id7 , :ol_quantity7 )
, ( SMALLINT( 9 ), :id8 , :ol_quantity8 )
, ( SMALLINT( 10 ), :id9 , :ol_quantity9 )
, ( SMALLINT( 11 ), :id10 , :ol_quantity10 )
, ( SMALLINT( 12 ), :id11 , :ol_quantity11 )
, ( SMALLINT( 13 ), :id12 , :ol_quantity12 )
, ( SMALLINT( 14 ), :id13 , :ol_quantity13 )
, ( SMALLINT( 15 ), :id14 , :ol_quantity14 )

) AS X ( OL_NUMBER , I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL

WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

)

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

FROM NEW TABLE ( INSERT INTO ORDER_LINE

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

INCLUDE( I_PRICE DECIMAL(5,2)

```

```

, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)

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

FROM DATA

) AS INS
;

// Start processing

in_newword = (struct in_newword_struct *) pin;
newword = (struct out_newword_struct *) pout;

#ifdef DEBUG
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++ )

```





```

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(NEWWORD_SQL, "Default switch on remote orderline/stock/index", __FILE__, __LINE__,
&sqlca);
goto ferror;
}
}
for ( inputItemArrayIndex = 0 ;
inputItemArrayIndex < in_newword->s_O_OL_CNT // from input
&& i_priceArray[ inputItemArrayIndex ] != 0 ;
inputItemArrayIndex++ )
{
// s_L_NAME, and s_S_QUANTITY already set as output host variables
newword->item[ inputItemArrayIndex ].s_L_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 )
)
{
newword->item[ inputItemArrayIndex ].s_brand_generic = 'B';
}
else
{
newword->item[ inputItemArrayIndex ].s_brand_generic = 'G';
}
}
EXEC SQL
SELECT W_TAX, C_DISCOUNT, C_LAST, C_CREDIT, O_ENTRY_D
INTO :ware_tax, :c_discount, :c_last, :c_credit, :o_entry_d

```

```

FROM TABLE ( NEW_WH ( :next_o_id
, :w_id
, :d_id
, :c_id
, :inputItemCount
, :allLocal
)
) AS NEW_WH_TABLE
;
if ( sqlca.sqlcode == 0 )
{
if ( newword->s_O_OL_CNT == in_newword->s_O_OL_CNT )
{
newword->s_transtatus = TRAN_OK ;
EXEC SQL COMMIT ;
if( sqlca.sqlcode != 0 )
{
sqlerror(NEWWORD_SQL, "COMMIT", __FILE__, __LINE__, &sqlca) ;
goto ferror;
}
}
else
{
newword->s_transtatus = INVALID_ITEM ;
EXEC SQL ROLLBACK WORK ;
if ( sqlca.sqlcode != 0 )
{
newword->s_transtatus = FATAL_SQLERROR;
sqlerror(NEWWORD_SQL, "ROLLBACK FAILED (INVALID ITEM)", __FILE__, __LINE__, &sqlca);
// no point in ferror
}
}
}
else
{
DLCHK( retry_tran );
sqlerror( NEWWORD_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 DEBUG
new_debug( newword, in_newword, "SP prior to return");
#endif
return ( storedProcRc ) ;
sql_error:
{
char tempstr[ 4096 ] ;

```

```

DLCHK( retry_tran ) ;
sprintf( tempstr, "inputItemCount=%d, :next_o_id=%d, :d_id=%d, :w_id=%d", inputItemCount,
next_o_id, d_id, w_id ) ;
sqlerror( NEWWORD_SQL, tempstr , __FILE__, __LINE__, &sqlca ) ;
}
}
ferror:
newword->s_transtatus = FATAL_SQLERROR;
EXEC SQL ROLLBACK WORK;
if ( sqlca.sqlcode != 0 )
{
sqlerror( NEWWORD_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*/
8,8,8,8,1,8,8,8,8, /*60-69*/
8,4,8,3,8,8,0,8,2,7, /*70-79*/
8,8,8,8,8,8,8,8,8, /*80-89*/
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 DEBUG
    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 ) ;

error:

    ordstat->s_transtatus = FATAL_SQLERROR ;

    EXEC SQL ROLLBACK WORK ;

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

    goto mexit;
}

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

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

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

    struct sqlca sqlca ;

    int storedProcRc ;

    short district_id ;
    sqlint32 customer_id ;

    EXEC SQL BEGIN DECLARE SECTION;

        // input

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

        // output

        short no_o_id_indicator = 0 ;
        sqlint32 no_o_id ;

    EXEC SQL END DECLARE SECTION;

    #define d_id district_id
    #define c_id customer_id

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

```

```

delivery->deadlocks = -1 ;

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

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

d_id = 1;

retry_tran:

    delivery->deadlocks++;

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

        EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

            SELECT O_ID

                INTO :no_o_id :no_o_id_indicator

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

        COMMIT ;

        END COMPOUND ;

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

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

    delivery->s_transtatus = TRAN_OK ;

mexit:

    if ( sqlca.sqlcode >= 0 )
    {

```

```

        storedProcRc = SQLZ_HOLD_PROC ;
    }
    else
    {
        storedProcRc = SQLZ_DISCONNECT_PROC ;
    }
}

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

    return ( storedProcRc ) ;

error:

    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 ) ;
}

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
*
*/

#ifdef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

```

```

#include "ival.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_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_ID 3849
#define A_C_LAST 255
#define A_C_ID 1023
#define A_OL_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_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 oiitems_struct {
        double s_OL_AMOUNT;
        int32_t s_OL_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_OL_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 20100315.1027 */
```

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 104040
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H
```

## include/platform.h

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

```
/*
 * platform.h - Platform Isolation Layer
 */
```

```
#ifndef __PLATFORM_H
#define __PLATFORM_H
```

```
/* *****
/* Generic Macros */
/* *****
#define GEN_ERRCODE errno
```

```
/* *****
/* Windows I/O Macros */
/* *****
/* *****
/* UNIX I/O Macros */
/* *****
#include <fcntl.h>

#define IOH_INIT(hnd, type, name)
hnd->fd = -1;
hnd->type = type;
hnd->name = name;

#define IOH_CREATE(hnd)
if (hnd->type == IOH_PIPE) {
rc = mkfifo(hnd->name, 0666);
} else {
rc = 0;
}

#define IOH_OPEN(hnd)
if (hnd->type == IOH_FILE_APPEND) {
hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_APPEND, 0666);
} else {
hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_TRUNC, 0666);
}
if (hnd->fd == -1) {
rc = -1;
} else {
rc = 0;
}

#define IOH_WRITE(hnd, buff, num, num2)
rc = write(hnd->fd, buff, num);
if (rc >= 0) {
num2 = rc;
rc = 0;
}

#define IOH_FLUSH(hnd) rc = 0;
#define IOH_CLOSE(hnd) rc = close(hnd->fd);
#define IOH_DELETE(hnd) if (hnd->type == IOH_PIPE) { rc = unlink(hnd->name); }
```

```
typedef unsigned int IOH_NUM;
typedef int IOH_HND;
```

```
/* *****
/* UNIX Semaphore Macros */
/* *****
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
```

```
union semun {
int val;
struct semid_ds *buf;
unsigned short int *array;
} semUnion;
```

```
struct sembuf semBuf;
```

```
#define SEM_HANDLE int
```

```
#define SEM_INIT(hnd, x, name)
if ( (hnd = semget(IPC_PRIVATE, 1, IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH)) == -1)
API_ERROR(__LINE__, "semget", (rc=GEN_ERRCODE));
semUnion.val = x;
if ( semctl(hnd, 0, SETVAL, semUnion) < 0 )
API_ERROR(__LINE__, "semctl SETVAL", (rc=GEN_ERRCODE));
```

```
#define SEM_WAIT(hnd)
```

```
semBuf.sem_num = 0;
semBuf.sem_op = -1;
semBuf.sem_flg = SEM_UNDO;
if ( semop(hnd, &semBuf, 1) < 0 )
API_ERROR(__LINE__, "semop wait", (rc=GEN_ERRCODE));
```

```
#define SEM_FREE(hnd)
semBuf.sem_num = 0;
semBuf.sem_op = 1;
semBuf.sem_flg = SEM_UNDO;
if ( semop(hnd, &semBuf, 1) < 0 )
API_ERROR(__LINE__, "semop free", (rc=GEN_ERRCODE));
```

```
#define SEM_DESTROY(hnd)
if ( semctl(hnd, 0, IPC_RMID, 0) )
API_ERROR(__LINE__, "semctl IPC_RMID", (rc=GEN_ERRCODE));
```

```
/* *****
/* Common I/O Macros and Definitions */
/* *****
```

```
#define IOH_FILE 1
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3
```

```
#define IOH_ERRMSG(hnd, msg)
if (rc != 0) {
fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", GEN_ERRCODE, msg,
hnd->fd, hnd->type, hnd->name);
return rc;
}
```

```
struct _ioh {
IOH_HND fd;
int type;
char *name;
};
```

```
typedef struct _ioh ioHandle;
```

```
/* *****
/* Generic I/O Routine Prototypes */
/* *****
```

```
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes);
int GenericClose(ioHandle *hnd);
```

```
/* *****
/* Generic I/O Routines */
/* *****
```

```
int GenericOpen(ioHandle *hnd, int type, char *name)
{
int rc = 0;
```

```
IOH_INIT(hnd, type, name)
```

```
IOH_CREATE(hnd)
IOH_ERRMSG(hnd, "creating")
```

```
IOH_OPEN(hnd)
IOH_ERRMSG(hnd, "opening")
```

```
return rc;
```

```
}
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes)
{
int rc = 0;
int numBytesWritten = -1;
```

```
IOH_WRITE(hnd, Buffer, numBytes, numBytesWritten)
IOH_ERRMSG(hnd, "writing")
if (numBytes != numBytesWritten) {
```

```

    fprintf(stderr, "Truncated data writing to fd %d (%d, %s)n", hnd->fd, hnd->type, hnd->name);
    rc = -1;
}
return rc;
}

int GenericClose(ioHandle *hnd)
{
    int rc = 0;

    IOH_FLUSH(hnd)
    IOH_ERRMSG(hnd, "flushing")

    IOH_CLOSE(hnd)
    IOH_ERRMSG(hnd, "closing")

    IOH_DELETE(hnd)
    IOH_ERRMSG(hnd, "deleting")

    return rc;
}

#endif // __PLATFORM_H

```

## include/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 - 2005
** 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"

```

```

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

```

```

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

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

        *a ^= *b;
        *b ^= *a;
        *a ^= *b;
    }
}

#endif //SWAP_ENDIAN

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

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

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

#endif // __TPCCAPP_H

```

## include/tpccdbg.h

```

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

```

```

/*
 * tpccdbg.h - Debugging Macros
 */

```

```

#ifndef __TPCCDBG_H
#define __TPCCDBG_H

```

```

#ifdef __cplusplus
extern "C" {
#endif

```

```

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

```

```

extern void new_debug (struct out_neword_struct *neword_ptr,
                    struct in_neword_struct *in_neword_ptr,
                    char *msg);

```

```

extern void pay_debug (struct out_payment_struct *payment_ptr,
                    struct in_payment_struct *in_payment_ptr,
                    char *msg);

```

```

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

```

## 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 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=CK080131

# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}

# The OS being used (i.e. "UNIX", "LINUX", "WINDOWS")
export PLATFORM=UNIX
export SERVER_PLATFORM=UNIX

# 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=v9

# The schema name is typically the SQL authorization ID (or username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}
export SERVER_TPCC_SCHEMA=${USER}

# DB2 EE/EEE Configuration
export DB2EDITION=EEE
#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

```

## tpccCom/compreg.h

```

// compreg.h : Declaration of the CCompReg

#pragma once

#include "resource.h" // main symbols
#include "tpccCom.h"

// CCompReg
class ATL_NO_VTABLE CCompReg :
public CComObjectRootEx<CComSingleThreadModel>,
public CComCoClass<CCompReg, &CLSID_CompReg>,
public IDispatchImpl<IComponentRegistrar, &IID_IComponentRegistrar, &LIBID_tpccComLib,
/*wMajor =*/ 1, /*wMinor =*/ 0>
{
public:
    CCompReg()
    {
    }

DECLARE_NO_REGISTRY()

BEGIN_COM_MAP(CCompReg)
    COM_INTERFACE_ENTRY(IComponentRegistrar)
    COM_INTERFACE_ENTRY(IDispatch)
END_COM_MAP()

// IComponentRegistrar
public:
    STDMETHOD(Attach)(BSTR bstrPath)
    {
        return S_OK;
    }
    STDMETHOD(RegisterAll)()
    {
        return _AtlComModule.RegisterServer(TRUE);
    }
    STDMETHOD(UnregisterAll)()
    {
        _AtlComModule.UnregisterServer(TRUE);
        return S_OK;
    }
    STDMETHOD(GetComponents)(SAFEARRAY *ppCLSIDs, SAFEARRAY *ppDescriptions)
    {
        if (ppCLSIDs == NULL || ppDescriptions == NULL )
            return E_POINTER;
        int nComponents = 0;
        for (_ATL_OBJMAP_ENTRY** ppEntry = _AtlComModule.m_ppAutoObjMapFirst; ppEntry
< _AtlComModule.m_ppAutoObjMapLast; ppEntry++)
        {
            if (*ppEntry != NULL)
            {
                _ATL_OBJMAP_ENTRY* pEntry = *ppEntry;
                if (pEntry->pclsid != NULL)

```

```

                LPCTSTR pszDescription = pEntry->pfnGetObjectDescription();
                if (pszDescription)
                    nComponents++;
            }
        }
    }
    SAFEARRAYBOUND rgBound{1};
    rgBound[0].lbound = 0;
    rgBound[0].cElements = nComponents;
    *ppCLSIDs = SafeArrayCreate(VT_BSTR, 1, rgBound);
    if (*ppCLSIDs == NULL )
        return AtlHresultFromLastError();
    *ppDescriptions = SafeArrayCreate(VT_BSTR, 1, rgBound);
    if (*ppDescriptions == NULL )
        return AtlHresultFromLastError();
    LONG i = 0;
    for (_ATL_OBJMAP_ENTRY** ppEntry = _AtlComModule.m_ppAutoObjMapFirst; ppEntry
< _AtlComModule.m_ppAutoObjMapLast; ppEntry++)
    {
        if (*ppEntry != NULL)
        {
            _ATL_OBJMAP_ENTRY* pEntry = *ppEntry;
            if (pEntry->pclsid != NULL)
            {
                LPCTSTR pszDescription = pEntry->pfnGetObjectDescription();
                if (pszDescription)
                {
                    LPOLESTR pszCLSID;
                    StringFromCLSID(*pEntry->pclsid, &pszCLSID);
                    BSTR pBSTR = OLE2BSTR(pszCLSID);
                    if (pBSTR == NULL )
                    {
                        CoTaskMemFree(pszCLSID);
                        return E_OUTOFMEMORY;
                    }
                    HRESULT hResult = SafeArrayPutElement(*ppCLSIDs,
                    &i, pBSTR);
                    CoTaskMemFree(pszCLSID);
                    if (FAILED(hResult) )
                        return hResult;
                    pBSTR = T2BSTR_EX(pszDescription);
                    if (pBSTR == NULL )
                    {
                        return E_OUTOFMEMORY;
                    }
                    hResult = SafeArrayPutElement(*ppDescriptions, &i,
                    pBSTR);
                    if (FAILED(hResult) )
                        return hResult;
                    i++;
                }
            }
        }
    }
    return S_OK;
}
STDMETHOD(RegisterComponent)(BSTR bstrCLSID)
{
    CLSID clsid;
    CLSIDFromString(bstrCLSID, &clsid);
    _AtlComModule.RegisterServer(TRUE, &clsid);
    return S_OK;
}
STDMETHOD(UnregisterComponent)(BSTR bstrCLSID)
{
    CLSID clsid;
    CLSIDFromString(bstrCLSID, &clsid);
    _AtlComModule.UnregisterServer(FALSE, &clsid);
    return S_OK;
}
};

```

```
OBJECT_ENTRY_AUTO(CLSID_CompReg, CCompReg)
```

## tpccCom/compreg.cpp

```
// compreg.cpp : Implementation of CCompReg
```

```
#include "stdafx.h"  
#include "compreg.h"
```

```
// CCompReg
```

## tpccCom/stdafx.h

```
// stdafx.h : include file for standard system include files,  
// or project specific include files that are used frequently,  
// but are changed infrequently
```

```
#pragma once
```

```
#ifndef STRICT  
#define STRICT  
#endif
```

```
// Modify the following defines if you have to target a platform prior to the ones specified below.  
// Refer to MSDN for the latest info on corresponding values for different platforms.
```

```
#ifndef WINVER // Allow use of features specific to Windows 95 and Windows NT  
4 or later.  
#define WINVER 0x0400 // Change this to the appropriate value to target Windows 98 and  
Windows 2000 or later.  
#endif
```

```
#ifndef _WIN32_WINNT // Allow use of features specific to Windows NT 4.0 or later.  
#define _WIN32_WINNT 0x0400 // Change this to the appropriate value to target Windows 2000 or later.  
#endif
```

```
#ifndef _WIN32_WINDOWS // Allow use of features specific to Windows 98 or later.  
#define _WIN32_WINDOWS 0x0410 // Change this to the appropriate value to target Windows Me or later.  
#endif
```

```
#ifndef _WIN32_IE // Allow use of features specific to IE 4.0 or later.  
#define _WIN32_IE 0x0400 // Change this to the appropriate value to target IE 5.0 or later.  
#endif
```

```
#define _ATL_APARTMENT_THREADED  
#define _ATL_NO_AUTOMATIC_NAMESPACE
```

```
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS // some CString constructors will be explicit
```

```
// turns off ATL's hiding of some common and often safely ignored warning messages  
#define _ATL_ALL_WARNINGS
```

```
#include <comsvcs.h>
```

```
#include "resource.h"  
#include <atlbase.h>  
#include <atlcom.h>
```

```
using namespace ATL;
```

## tpccCom/stdafx.cpp

```
// stdafx.cpp : source file that includes just the standard includes  
// tpccCom.pch will be the pre-compiled header  
// stdafx.obj will contain the pre-compiled type information
```

```
#include "stdafx.h"
```

## tpccCom/dlldata.x.h

```
#pragma once
```

```
#ifndef _MERGE_PROXYSTUB
```

```
extern "C"  
{  
    BOOL WINAPI PrxDllMain(HINSTANCE hinstance, DWORD dwReason,  
        LPVOID lpReserved);  
    STDAPI PrxDllCanUnloadNow(void);  
    STDAPI PrxDllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv);  
    STDAPI PrxDllRegisterServer(void);  
    STDAPI PrxDllUnregisterServer(void);  
}
```

```
#endif
```

## tpccCom/Resource.h

```
//{{NO_DEPENDENCIES}  
// Microsoft Visual C++ generated include file.  
// Used by tpccCom.rc
```

```
//  
#define IDS_PROJNAME 100  
#define IDR_TPCCCOM 101  
#define IDR_TPCC_COM 102
```

```
// Next default values for new objects
```

```
//  
#ifndef APSTUDIO_INVOKED  
#ifndef APSTUDIO_READONLY_SYMBOLS  
#define _APS_NEXT_RESOURCE_VALUE 201  
#define _APS_NEXT_COMMAND_VALUE 32768  
#define _APS_NEXT_CONTROL_VALUE 201  
#define _APS_NEXT_SYMED_VALUE 103  
#endif  
#endif
```

## tpccCom/tpccCom.idl

```
// tpccCom.idl : IDL source for tpccCom  
//
```

```
// This file will be processed by the MIDL tool to  
// produce the type library (tpccCom.tlb) and marshalling code.
```

```
import "oaidl.idl";  
import "ocidl.idl";  
//this is test.
```

```
{  
    object,  
    uuid(a817e7a2-43fa-11d0-9e44-00aa00b6770a),  
    dual,  
    helpstring("IComponentRegistrar Interface"),  
    pointer_default(unique)  
}  
interface IComponentRegistrar : IDispatch  
{  
    [id(1)] HRESULT Attach([in] BSTR bstrPath);  
    [id(2)] HRESULT RegisterAll();  
    [id(3)] HRESULT UnregisterAll();  
    [id(4)] HRESULT GetComponents([out] SAFEARRAY(BSTR)* pbsrCLSIDs, [out]  
SAFEARRAY(BSTR)* pbsrDescriptions);  
    [id(5)] HRESULT RegisterComponent([in] BSTR bstrCLSID);  
    [id(6)] HRESULT UnregisterComponent([in] BSTR bstrCLSID);  
};
```

```
[  
    object,  
    uuid(5B4FA473-2E68-4D79-A626-F38B30B8196E),  
    helpstring("Itpcc_com Interface"),  
    pointer_default(unique)  
]  
interface Itpcc_com : IUnknown{  
    [helpstring("method doStockLevel")] HRESULT doStockLevel([in] INT* size, [in,out, size_is(*size)]  
    UCHAR**buffer);  
    [helpstring("method doNewOrder")] HRESULT doNewOrder([in] INT* size, [in,out, size_is(*size)]  
    UCHAR** buffer);  
    [helpstring("method doPayment")] HRESULT doPayment([in] INT* size, [in,out, size_is(*size)]  
    UCHAR** buffer);  
    [helpstring("method doOrderStatus")] HRESULT doOrderStatus([in] INT* size,  
[in,out, size_is(*size)] UCHAR** buffer);  
    [helpstring("method doDBInfo")] HRESULT doDBInfo(void);  
    [helpstring("method doSetComplete")] HRESULT doSetComplete(void);  
};  
[  
    uuid(91F1B8B0-89E9-457B-A228-3E2D6CE3E752),  
    version(1.0),  
    helpstring("tpccCom 1.0 Type Library"),  
    custom(a817e7a1-43fa-11d0-9e44-00aa00b6770a, "90EEDAFF-F8D3-4711-99A9-  
8AC3C0FE5DB9")  
]  
library tpccComLib  
{  
    importlib("stdole2.tlb");  
  
    [  
        uuid(90EEDAFF-F8D3-4711-99A9-8AC3C0FE5DB9),  
        helpstring("ComponentRegistrar Class")  
    ]  
    coclass CompReg  
    {  
        [default] interface IComponentRegistrar;  
    };  
    [  
        uuid(5F752BF2-F739-43D4-8492-44C19581C0A1),  
        helpstring("tpcc_com Class")  
    ]  
    coclass tpcc_com  
    {  
        [default] interface Itpcc_com;  
    };  
};
```

## tpccCom/dlldata.c

```
/*  
*****  
DllData file -- generated by MIDL compiler  
*****  
*/
```

```
DO NOT ALTER THIS FILE
```

```
This file is regenerated by MIDL on every IDL file compile.
```

```
To completely reconstruct this file, delete it and rerun MIDL  
on all the IDL files in this DLL, specifying this file for the  
/dlldata command line option
```

```
*****  
*/
```

```
#define PROXY_DELEGATION
```

```
#include <rpcproxy.h>
```

```
#ifdef _cplusplus  
extern "C" {  
#endif
```

```
EXTERN_PROXY_FILE( tpccCom )
```



```

PROXYFILE_LIST_START
/* Start of list*/
REFERENCE_PROXY_FILE( tpccCom ),
/* End of list */
PROXYFILE_LIST_END

```

```

DLldata_ROUTINES( aProxyFileList, GET_DLL_CLSID )

```

```

#ifdef __cplusplus
} /*extern "C"*/
#endif

```

```

/* end of generated dlldata file */

```

## tpccCom/dlldata.c

```

// wrapper for dlldata.c

```

```

#ifdef _MERGE_PROXYSTUB // merge proxy stub DLL

```

```

#define REGISTER_PROXY_DLL //DllRegisterServer, etc.

```

```

#define _WIN32_WINNT 0x0500//for Win2000, change it to 0x0400 for NT4 or Win95 with DCOM
#define USE_STUBLESS_PROXY //defined only with MIDL switch /Oicf

```

```

#pragma comment(lib, "rpcns4.lib")
#pragma comment(lib, "rpcrt4.lib")

```

```

#define ENTRY_PREFIX Prx

```

```

#include "dlldata.c"
#include "tpccCom_p.c"

```

```

#endif // _MERGE_PROXYSTUB

```

## tpccCom/tpccCom.cpp

```

// tpccCom.cpp : Implementation of DLL Exports.

```

```

//
// Note: COM+ 1.0 Information:
// Please remember to run Microsoft Transaction Explorer to install the component(s).
// Registration is not done by default.

```

```

#include "stdafx.h"
#include "resource.h"
#include "tpccCom.h"
#include "compreg.h"
#include "dlldata.h"
#include <string.h>

```

```

class CtpccComModule : public CAtdDllModuleT< CtpccComModule >
{
public :
    DECLARE_LIBID(LIBID_tpccComLib)
    DECLARE_REGISTRY_APPID_RESOURCEID(IDR_TPCCCOM, "{11ED2355-1A27-42F1-ADFF-
F201F5E82BCE}")
};

```

```

CtpccComModule _AtDllModule;

```

```

void handle_exit();

```

```

// DLL Entry Point
extern "C" BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved)
{
#ifdef _MERGE_PROXYSTUB

```

```

if (!PrxDllMain(hInstance, dwReason, lpReserved))
return FALSE;
#endif
hInstance;
atexit(handle_exit);
return _AtDllModule.DllMain(dwReason, lpReserved);
}

```

```

// Used to determine whether the DLL can be unloaded by OLE
STDAPI DllCanUnloadNow(void)

```

```

{
#ifdef _MERGE_PROXYSTUB
HRESULT hr = PrxDllCanUnloadNow();
if (FAILED(hr))
return hr;
#endif
return _AtDllModule.DllCanUnloadNow();
}

```

```

// Returns a class factory to create an object of the requested type
STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)

```

```

{
#ifdef _MERGE_PROXYSTUB
if (PrxDllGetClassObject(rclsid, riid, ppv) == S_OK)
return S_OK;
#endif
return _AtDllModule.DllGetClassObject(rclsid, riid, ppv);
}

```

```

// DllRegisterServer - Adds entries to the system registry
STDAPI DllRegisterServer(void)

```

```

{
// registers object, typelib and all interfaces in typelib
HRESULT hr = _AtDllModule.DllRegisterServer();
#ifdef _MERGE_PROXYSTUB
if (FAILED(hr))
return hr;
hr = PrxDllRegisterServer();
#endif
return hr;
}

```

```

// DllUnregisterServer - Removes entries from the system registry
STDAPI DllUnregisterServer(void)

```

```

{
HRESULT hr = _AtDllModule.DllUnregisterServer();
#ifdef _MERGE_PROXYSTUB
if (FAILED(hr))
return hr;
hr = PrxDllUnregisterServer();
if (FAILED(hr))
return hr;
hr = PrxDllUnregisterServer();
#endif
return hr;
}

```

```

// DllInstall - Adds/Removes entries to the system registry per user
// per machine.
STDAPI DllInstall(BOOL bInstall, LPCWSTR pszCmdLine)

```

```

{
HRESULT hr = E_FAIL;
static const wchar_t szUserSwitch[] = L"user";

if (pszCmdLine != NULL)
{
if (_wcsnicmp(pszCmdLine, szUserSwitch, _countof(szUserSwitch)) == 0)
{
AtSetPerUserRegistration(true);

```

```

}
}
if (bInstall)
{
hr = DllRegisterServer();
if (FAILED(hr))
{
DllUnregisterServer();
}
}
else
{
hr = DllUnregisterServer();
}
return hr;
}

```

## tpccCom/tpcc\_com.h

```

// tpcc_com.h : Declaration of the Ctpcc_com

```

```

#pragma once
#include "tpccCom.h"
#include "resource.h" // main symbols
#include <comsvcs.h>

```

```

#include ".\tpccsapitpcc.h"

```

```

#ifdef WIN32
extern struct _timeb start_time; // time that the shared memory was initialized
#else
extern struct timeb start_time; // time that the shared memory was initialized
#endif

```

```

#ifdef ORACLE
#define ORA_NT
#include <stdio.h>
#include <time.h>
#include <ora_tpcc.h>
#include <plora.h>
#endif

```

```

#define NULL_DB "nullDB"

static HINSTANCE dbInstance = NULL;

```

```

static CRITICAL_SECTION debugMutex;
static CRITICAL_SECTION errorMutex;
static ofstream debugStream;
static ofstream errorStream;

```

```

static int comServerID = 0;
static int debugFileOpen = 0;
static int errorFileOpen = 0;
static int nullDB = 0;
static char dbType[32];
static char dbName[32];

```

```

typedef INT (*NORD_PTR)(nord_wrapper *nord,void *connectHandle);
typedef INT (*PYMT_PTR)(paym_wrapper *pymt,void *connectHandle);
typedef INT (*ORDS_PTR)(ords_wrapper *ords,void *connectHandle);
typedef INT (*STOK_PTR)(stok_wrapper *stok,void *connectHandle);
typedef INT (*CONNECT_PTR)(char *dbName,void **connectHandle);
typedef INT (*DISCONNECT_PTR)(void *connectHandle);
typedef INT (*DBINIT_PTR)(void *connectHandle);
DBINIT_PTR DBInit;

```

```

NORD_PTR do_nord;

```

```

PYMT_PTR do_pymt;
ORDS_PTR do_ords;
STOK_PTR do_stok;
CONNECT_PTR do_connection;
DISCONNECT_PTR do_disconnect;

// Ctpcc_com
class ATL_NO_VTABLE Ctpcc_com :
public CComObjectRootEx<CComMultiThreadModel>,
public IObjectControl,
public CComCoClass<Ctpcc_com, &CLSID_tpcc_com>,
public Itpcc_com
{
public:
    Ctpcc_com()
    {
        int rc = ERR;
        connected = 0;
        connectHandleInUse = 0;
        bool again = false;
        int count = 0;

        if(debugFlag)
        {
            if(!debugFileOpen)
            {
                InitializeCriticalSection(&debugMutex);
                //open comLog
                char comLogFile[128];
                sprintf(comLogFile, "C:\\inetpub\\wwwroot\\tpcc\\comLog_debug.txt");
                debugStream.rdbuf( >->open(comLogFile, ios_base::in | ios_base::out |
ios_base::app);

                debugFileOpen = 1;
                DEBUGMSG("Debug file open for writing"<<endl);
            }
        }

        //open error log file
        if(!errorFileOpen)
        {
            InitializeCriticalSection(&errorMutex);
            char errorLogFile[128];
            sprintf(errorLogFile, "C:\\inetpub\\wwwroot\\tpcc\\comLog_err.txt");
            errorStream.rdbuf( >->open(errorLogFile, ios_base::in | ios_base::out );
            errorFileOpen=1;
        }
#ifdef RT_DEBUG
#ifdef WIN32
        _ftime(&start_time); // get the time when shared memory is created
#else
        ftime(&start_time); // get the time when shared memory is created
#endif
#endif

        //get registry values
        if((rc = readRegistry()) != OK)
        {
            rc:" << rc <<endl);
            ERRORMSG("Unable to open registry key " << REGISTRY_SUB_KEY << "
rc:" << rc <<endl);
            DEBUGMSG("Unable to open registry key " << REGISTRY_SUB_KEY << "
rc:" << rc <<endl);
            return;
        }
        DEBUGMSG("nullDB:" <<nullDB<<" dbType:"<<dbType<<" dbName:"<<dbName<<endl);

        //load library based on registry
        if( (rc = loadLibrary()) != OK)
        {

```

```

            ERRORMSG("load library failure rc:" << rc << endl);
            return;
        }
    }

    DEBUGMSG("dbtype:"<<dbType<<" instance:" << DEBUGADDRESS(dbInstance) << "
loaded." << endl);

    //connect to db
    do {
        again = false;
        EnterCriticalSection(&errorMutex);
        if((rc = connectDB()) != OK)
        {
            ERRORMSG("unable to connect to db "<<dbName<<" rc:"<<rc <<endl);
            LeaveCriticalSection(&errorMutex);
            if(count++ > 5)
                return;
            Sleep(5000);
        }
    } while (again);
    LeaveCriticalSection(&errorMutex);

    DEBUGMSG("connected to db " <<dbName<<" rc:"<< rc << " context:"
<<DEBUGADDRESS(connectHandle) << endl);
}

DECLARE_PROTECT_FINAL_CONSTRUCT()

HRESULT FinalConstruct()
{
    return S_OK;
}

void FinalRelease()
{
}

DECLARE_REGISTRY_RESOURCEID(IDR_TPCC_COM)

BEGIN_COM_MAP(Ctpcc_com)
    COM_INTERFACE_ENTRY(Itpcc_com)
    COM_INTERFACE_ENTRY(IObjectControl)
END_COM_MAP()

// IObjectControl
public:
    STDMETHODCALLTYPE(Activate)();
    STDMETHODCALLTYPE(BOOL, CanBePooled)();
    STDMETHODCALLTYPE(void, Deactivate)();
    CComPtr<ObjectContext> m_spObjectContext;

// Itpcc_com
public:
    STDMETHODCALLTYPE(doStockLevel)(INT *size, UCHAR **buffer);
    STDMETHODCALLTYPE(doNewOrder)(INT *size, UCHAR** buffer);
    STDMETHODCALLTYPE(doPayment)(INT *size, UCHAR** buffer);
    STDMETHODCALLTYPE(doOrderStatus)(INT *size, UCHAR** buffer);
    STDMETHODCALLTYPE(doDBInfo)(void);
    STDMETHODCALLTYPE(doSetComplete)(void);

    int connected;
    int connectHandleInUse;

private:
    void *connectHandle;
    int loadLibrary();
    int readRegistry();
    int connectDB();
    void handle_exit();
};

```

```

OBJECT_ENTRY_AUTO(__uuidof(tpcc_com), Ctpcc_com)

tpccCom/tpcc_com.cpp
// tpcc_com.cpp : Implementation of Ctpcc_com

#include "stdafx.h"
#include "tpcc_com.h"

#include "..\tpccsapitpcc.h"

#include "..\tpcc_com.h"
#ifdef DB2
#include <db2tpcc.h>
#endif
#ifdef ORA
#include <ora_tpcc.h>
#endif

#ifdef DEBUG
    int debugFlag = 1;
#else
    int debugFlag = 0;
#endif

#ifdef RT_DEBUG
#ifdef WIN32
    struct _timeb start_time; // time that the shared memory was initialized
#else
    struct timeb start_time; // time that the shared memory was initialized
#endif
#endif

#ifdef RT_DEBUG
//
=====
UINT timestamp()
{
#ifdef WIN32
    struct _timeb tb;
    _ftime( &tb );
#else
    struct timeb tb;
    ftime( &tb );
#endif
// returns time since the SharedMem class was constructed
return ((tb.time - start_time.time) * 1000) + tb.millitm - start_time.millitm;
}
#endif

// Ctpcc_com
void handle_exit()
{
    ERRORMSG("Exit caught"<<endl);
    DBInit(NULL);
}

HRESULT Ctpcc_com::Activate()
{
    HRESULT hr = GetObjectContext(&m_spObjectContext);
    if (SUCCEEDED(hr))
    {
        DEBUGMSG("Object assigned to thread."<<endl);
        return S_OK;
    }
    return hr;
}

BOOL Ctpcc_com::CanBePooled()
{

```

```

        DEBUGMSG("CanBePooled() returning true"<<endl);
        return TRUE;
    }

void Ctpcc_com::Deactivate()
{
    DEBUGMSG("deactivated() releasing object back into pool"<<endl);
    m_spObjectContext.Release();
}

/*
*****
** Name          : doSetComplete
** Description   : Release object back into com pool
** Parameters    :
** Returns       : int - return code
** Comments     : Calls SetComplete on the object that the com
                  pool manager returned to the caller(isapi thread)
*****
*/
STDMETHODIMP Ctpcc_com::doSetComplete(void)
{
    // TODO: Add your implementation code here
    HRESULT hres = m_spObjectContext->SetComplete();
    if (SUCCEEDED(hres))
    {
        DEBUGMSG("SetComplete successful. object bit set to release object into pool."<<endl);
    }
    else
    {
        DEBUGMSG("SetComplete failed. object bit set to release object into pool."<<endl);
        ERRORMSG("SetComplete() failed. code:"<<HRESULT_CODE(hres)<<"
        facility:"<<HRESULT_FACILITY(hres)<<" hres:"<<hex<<hres<<endl);
    }

    return S_OK;
}

/*
*****
** Name          : doStockLevel
** Description   : Call db2 dll entry point to execute txn
** Parameters    :
** Returns       : int*          size of UCHAR buffer to pay attention to
                  UCHAR**       char buffer that holds txn wrapper struct
** Comments     : int - return code
*****
*/
STDMETHODIMP Ctpcc_com::doStockLevel(INT *size, UCHAR **buffer)
{
    stok_wrapper *stok;
    stok = (stok_wrapper *) *buffer;

#ifdef RT_DEBUG
    struct _timeb times[4];
    _ftime(&times[0]);
#endif

    if(!connectHandleInUse)
    {
        DEBUGMSG("Setting Context handle in use to true"<<endl);
        connectHandleInUse = 1;
    }
    else
    {
        DEBUGMSG("Context handle in use."<<endl);
        ERRORMSG("Context handle in use."<<endl);
    }
}

```

```

        return ERR_HANDLE_IN_USE;
    }

    DEBUGMSG("Calling do_stok call using
    connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<stok->in_stok.s_W_ID<<" d_id:"<<
    stok->in_stok.s_D_ID<<
        " s_transtatus:"<<stok->out_stok.s_transtatus<<endl);

    int retry = 0;
    do
    {
        do_stok(stok,connectHandle);
    } while (stok->out_stok.s_transtatus && retry++ < 3);

    DEBUGMSG("Return from do_stok call using
    connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<stok->in_stok.s_W_ID<<" d_id:"<<
    stok->in_stok.s_D_ID<<
        " s_transtatus:"<<stok->out_stok.s_transtatus<<endl);

    DEBUGMSG("Connection handle set to free" <<endl);
    connectHandleInUse = 0;

#ifdef RT_DEBUG
    _ftime(&times[1]);
    ERRORMSG("3 - " << stok->in_stok.s_W_ID << " " << stok->in_stok.s_D_ID << " " << stok-
    >out_stok.s_low_stock << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " " <<
    times[1].millitm << endl);
#endif
    return S_OK;
}

/*
*****
** Name          : doNewOrder
** Description   : Call db2 dll entry point to execute txn
** Parameters    :
** Returns       : int*          size of UCHAR buffer to pay attention to
                  UCHAR**       char buffer that holds txn wrapper struct
** Comments     : int - return code
*****
*/
STDMETHODIMP Ctpcc_com::doNewOrder(INT* size, UCHAR** buffer)
{
    char n[818+264];
    nord_wrapper *nord=(nord_wrapper *)&n;
    nord_wrapper *nord = (nord_wrapper *) *buffer;
    memcpy(nord,(void *) *buffer,sizeof(nord_wrapper));

#ifdef RT_DEBUG
    struct _timeb times[4];
    _ftime(&times[0]);
#endif

    if(!connectHandleInUse)
    {
        DEBUGMSG("Setting Context handle in use to true"<<endl);
        connectHandleInUse = 1;
    }
    else
    {
        DEBUGMSG("Context handle in use."<<endl);
        ERRORMSG("Context handle in use."<<endl);
        return ERR_HANDLE_IN_USE;
    }

    DEBUGMSG("Calling do_nord call using
    connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<nord->in_nord.s_W_ID<<" d_id:"<<
    nord->in_nord.s_D_ID<<
        " s_transtatus:"<<nord->out_nord.s_transtatus<<endl);

    do_nord(nord,connectHandle);
}

```

```

// if (nord->out_nord.s_transtatus)
// {
//     ERRORMSG("Transaction error" <<endl);
// }
// DEBUGMSG("Return from do_nord call using
// connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<nord->in_nord.s_W_ID<<" d_id:"<<
// nord->in_nord.s_D_ID<<
//     " s_transtatus:"<<nord->out_nord.s_transtatus<<endl);
// try
// {
//     memcpy((void *) *buffer,nord,sizeof(nord_wrapper));
// }
// catch (...)
// {
//     ERRORMSG("Connection handle set to free" <<endl);
// }
// DEBUGMSG("Connection handle set to free" <<endl);
// connectHandleInUse = 0;

#ifdef RT_DEBUG
    _ftime(&times[1]);
    ERRORMSG("0 - " << nord->in_nord.s_W_ID << " " << nord->in_nord.s_D_ID << " " << nord-
    >out_nord.s_O_ID << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " " <<
    times[1].millitm << endl);
#endif
    return S_OK;
}

/*
*****
** Name          : doPayment
** Description   : Call db2 dll entry point to execute txn
** Parameters    :
** Returns       : int*          size of UCHAR buffer to pay attention to
                  UCHAR**       char buffer that holds txn wrapper struct
** Comments     : int - return code
*****
*/
STDMETHODIMP Ctpcc_com::doPayment(INT* size, UCHAR** buffer)
{
    paym_wrapper *pymt;
    pymt = (paym_wrapper *) *buffer;

#ifdef RT_DEBUG
    struct _timeb times[4];
    _ftime(&times[0]);
#endif

    if(!connectHandleInUse)
    {
        DEBUGMSG("Setting Context handle in use to true"<<endl);
        connectHandleInUse = 1;
    }
    else
    {
        DEBUGMSG("Context handle in use."<<endl);
        ERRORMSG("Context handle in use."<<endl);
        return ERR_HANDLE_IN_USE;
    }

    DEBUGMSG("Calling do_pymt call using
    connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<pymt->in_paym.s_W_ID<<" d_id:"<<
    pymt->in_paym.s_D_ID<<
        " s_transtatus:"<<pymt->out_paym.s_transtatus<<endl);

    do_pymt(pymt,connectHandle);
}

```

```

        DEBUGMSG("Return from do_pymt call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<pymt->in_paym.s_W_ID<<" d_id:"<<
pymt->in_paym.s_D_ID<<
        " s_transtatus:"<<pymt->out_paym.s_transtatus<<endl);

        DEBUGMSG("Connection handle set to free" <<endl);
connectHandleInUse = 0;

#ifdef RT_DEBUG
        ftime(&times[1]);
        ERRORMSG("1 - " << pymt->in_paym.s_W_ID << " " << pymt->in_paym.s_D_ID << " " << pymt-
>out_paym.s_C_BALANCE << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " "
<< times[1].millitm << endl);
#endif
        return S_OK;
}

/*
*****
** Name          : doOrderStatus
** Description   :
**              Call db2 dll entry point to execute txn
** Parameters   :
**              int*      size of UCHAR buffer to pay attention to
**              UCHAR**   char buffer that holds txn wrapper struct
** Returns      :
**              int - return code
** Comments     :
**
*****
*/

STDMETHODIMP Ctpcc_com::doOrderStatus(INT* size, UCHAR** buffer)
{
    ords_wrapper *ords;
    ords = (ords_wrapper *) *buffer;

#ifdef RT_DEBUG
    struct _timeb times[4];
    _ftime(&times[0]);
#endif

    if(!connectHandleInUse)
    {
        DEBUGMSG("Setting Context handle in use to true" <<endl);
        connectHandleInUse = 1;
    }
    else
    {
        DEBUGMSG("Context handle in use." <<endl);
        ERRORMSG("Context handle in use." <<endl);
        return ERR_HANDLE_IN_USE;
    }

    DEBUGMSG("Calling do_ords call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<ords->in_ords.s_W_ID<<" d_id:"<<
ords->in_ords.s_D_ID<<
        " s_transtatus:"<<ords->out_ords.s_transtatus<<endl);

    int retry = 0;
    do
    {
        do_ords(ords,connectHandle);
    } while (ords->out_ords.s_transtatus && retry++ < 3);

    DEBUGMSG("Return from do_ords call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<ords->in_ords.s_W_ID<<" d_id:"<<
ords->in_ords.s_D_ID<<
        " s_transtatus:"<<ords->out_ords.s_transtatus<<endl);

    DEBUGMSG("Connection handle set to free" <<endl);
connectHandleInUse = 0;

```

```

#ifdef RT_DEBUG
        ftime(&times[1]);
        ERRORMSG("2 - " << ords->in_ords.s_W_ID << " " << ords->in_ords.s_D_ID << " " << ords-
>out_ords.s_C_BALANCE << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " "
<< times[1].millitm << endl);
#endif
        return S_OK;
}

/*
*****
** Name          : doDBInfo
** Description   :
**              Function to test com interface
** Parameters   :
** Returns      :
**              int - return code
** Comments     :
**
*****
*/

STDMETHODIMP Ctpcc_com::doDBInfo(void)
{
    return S_OK;
}

/*
*****
** Name          : loadLibrary
** Description   :
**              Function loads appropiate db library based on
**              registry setting
** Parameters   :
** Returns      :
**              int - return code
** Comments     :
**
*****
*/

int Ctpcc_com::loadLibrary()
{
    ERRORMSG("Entered loadLibrary function" <<endl);
    //check to see if dbInstance is already loaded
    if(!dbInstance)
    {
        ERRORMSG("Database dll not loaded. Loading dll." <<endl);
        if (nullDB)
        {
            ERRORMSG("Loading "<<dbType <<" nulldb dll." << endl);
            dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\inuldb.dll");
            if(dbInstance == NULL)
            {
                DEBUGMSG("Unable to load null db dll, rc:"<<GetLastError());
                ERRORMSG("Unable to load null db dll, rc:"<<GetLastError());
                return ERR_NULL_DLL_NOT_LOADED;
            }
        }
        ERRORMSG(dbType <<" nulldb dll loaded" <<endl);
    }
    else if(strcmp(dbType,"DB2") == 0)
    {
        ERRORMSG("Loading "<<dbType <<" dll." << endl);

        dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccDB2glue.dll");
        if(dbInstance == NULL)
        {
            DEBUGMSG("Unable to load library." <<endl);

            ERRORMSG("Unable to load com dll, rc:" << hex << GetLastError()) <<

            return ERR_DB2_DLL_NOT_LOADED;
        }
    }
}

```

```

        }
        ERRORMSG(dbType <<" dll loaded" <<endl);
    }
    else if (strcmp(dbType,"ORACLE") == 0 )
    {
        ERRORMSG("Loading "<<dbType <<" dll." << endl);

        dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccOracleglue.dll");
        if(dbInstance == NULL)
        {
            DEBUGMSG("Unable to load oracle dll" <<endl);
            ERRORMSG("Unable to load oracle dll, rc:"<<GetLastError())<<endl);
            return ERR_ORACLE_DLL_NOT_LOADED;
        }
        ERRORMSG(dbType <<" dll loaded" <<endl);
    }
    else if (strcmp(dbType,"SYBASE") == 0 )
    {
        ERRORMSG("Loading "<<dbType <<" dll." << endl);

        dbInstance = LoadLibrary("c:\inetpub\wwwroot\tpcc\tpccSybaseGlue.dll");
        if(dbInstance == NULL)
        {
            DEBUGMSG("Unable to load sybase dll" <<endl);
            ERRORMSG("Unable to load sybase dll, rc:"<<GetLastError())<<endl);
            return ERR_ORACLE_DLL_NOT_LOADED;
        }
        ERRORMSG(dbType <<" dll loaded" <<endl);
    }
    else
    {
        DEBUGMSG("Unknown database type dll:"<<dbType<<endl);
        ERRORMSG("Unknown database type dll:"<<dbType<<endl);
        return ERR_UNKNOWN_DB;
    }

    //retrieve function addresses from instance loaded.
    DEBUGMSG("Getting do_connection function address from "<<dbType<<" dll" <<endl);
    if (do_connection = (CONNECT_PTR)GetProcAddress(dbInstance,"connect_db")) ==
NULL )
        return ERR_CONNECT_ADDRESS_NOT_FOUND;
    DEBUGMSG("do_connection address:"<<DEBUGADDRESS(do_connection)<<endl);

#ifdef DB2
    DEBUGMSG("Getting DBInit function address from "<<dbType<<" dll" <<endl);
    if (DBInit = (DBINIT_PTR)GetProcAddress(dbInstance,"DBInit")) == NULL )
        return ERR_DBINIT_ADDRESS_NOT_FOUND;
    DEBUGMSG("DBInit address:"<<DEBUGADDRESS(DBInit)<<endl);

#endif

#ifdef def
    DEBUGMSG("Getting do_nord function address from "<<dbType<<" dll" <<endl);
    if (do_nord = (NORD_PTR) GetProcAddress(dbInstance,"do_nord")) == NULL
        return ERR_NORD_ADDRESS_NOT_FOUND;
    DEBUGMSG("do_nord function address:"<<DEBUGADDRESS(do_nord)<<endl);

    DEBUGMSG("Getting do_pymt function address from "<<dbType<<" dll" <<endl);
    if (do_pymt = (PYMT_PTR) GetProcAddress(dbInstance,"do_pymt")) == NULL
        return ERR_PYMT_ADDRESS_NOT_FOUND;
    DEBUGMSG("do_pymt function address:"<<DEBUGADDRESS(do_pymt)<<endl);

    DEBUGMSG("Getting do_ords function address from "<<dbType<<" dll" <<endl);
    if (do_ords = (ORDS_PTR) GetProcAddress(dbInstance,"do_ords")) == NULL
        return ERR_ORDS_ADDRESS_NOT_FOUND;
    DEBUGMSG("do_ords function address:"<<DEBUGADDRESS(do_ords)<<endl);

    DEBUGMSG("Getting do_stok function address from "<<dbType<<" dll" <<endl);
    if (do_stok = (STOK_PTR) GetProcAddress(dbInstance,"do_stok")) == NULL
        return ERR_STOK_ADDRESS_NOT_FOUND;
    DEBUGMSG("do_stok function address:"<<DEBUGADDRESS(do_stok)<<endl);

    ERRORMSG("Getting do_disconnect function address from "<<dbType<<" dll" <<endl);
    if (do_disconnect = (DISCONNECT_PTR)GetProcAddress(dbInstance,"disconnect_db"))
== NULL )
        return ERR_DISCONNECT_ADDRESS_NOT_FOUND;

```

```

DEBUGMSG("do_disconnect address:"<<DEBUGADDRESS(do_disconnect)<<endl);

DEBUGMSG("All function addresses retrieved successfully."<<endl);

    }
    return OK;
}

/*
*****
** Name          :    readRegistry()
** Description   :
**              :    Function reads registry value
** Parameters   :
** Returns      :
**              :    int - return code
** Comments     :
**              :    Values retrieved from registry
**              :    dbName, dbUserName, and dbUserPassword
*****
*/

int Ctpcc_com::readRegistry()
{
    //open registry key
    HKEY registryKey;
    DWORD regType=NULL;
    char value[MAX_STRING_LEN];
    DWORD regValue;
    DWORD regValueSize = MAX_STRING_LEN;

    ERRORMSG("Entered readRegistry(), opening key:"<< REGISTRY_SUB_KEY <<endl);
    //open up registry key
    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTRY_SUB_KEY,0,KEY_QUERY_VALUE,&
registryKey) == ERROR_SUCCESS)
    {
        ERRORMSG(REGISTRY_SUB_KEY<<" open, getting database type from key"<<endl);
        regValueSize = sizeof(value);
        regType = REG_SZ;
        if (RegQueryValueEx(registryKey,TEXT("dbType"),NULL,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS ) {
            strcpy(dbType,value);
            ERRORMSG("Database type:"<<value<<" from registry key."<<endl);
        } else {
            ERRORMSG("Cannot read Database type from registry key."<<endl);
        }

        ERRORMSG("Getting database name from registry key."<<endl);
        regValueSize = sizeof(value);
        regType = REG_SZ;
        if (RegQueryValueEx(registryKey,DB_NAME,0,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS ) {
            strcpy(dbName,value);
            ERRORMSG("Database name:"<<dbName<<endl);
        } else {
            ERRORMSG("Cannot read Database name from registry key."<<endl);
        }

        ERRORMSG("Getting null database flag from key."<<endl);
        regValueSize = sizeof(regValue);
        regType = REG_DWORD;
        if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS ) {
            nullDB = regValue;
            ERRORMSG("Null database flag: "<<nullDB<<endl);
        } else {
            ERRORMSG("Cannot read nullDB from registry key."<<endl);
        }
        RegCloseKey(registryKey);
        return OK;
    }

    ERRORMSG("Error, unable to open registry key."<<endl);
    return ERR_UNABLE_TO_OPEN_REG;
}

```

```

}

/*
*****
** Name          :    connectDB
** Description   :
**              :    Function connects to the db
** Parameters   :
** Returns      :
**              :    int - return code
** Comments     :
**              :
*****
*/
int Ctpcc_com::connectDB()
{
    DEBUGMSG("Entered connectDB(), checking if object is connected."<<endl);
    if(!connected)
    {
        DEBUGMSG("Object not connected, calling do_connection with dbName:"<<dbName<<"
connectHandle:"<<
        DEBUGADDRESS(connectHandle)<<endl);
        if(!connectHandleInUse)
        {
            ERRORMSG("Setting Context handle in use to true"<<endl);
            connectHandleInUse = 1;
            ERRORMSG("connectHandle:"<<connectHandle<<endl);
            connected = do_connection(dbName,&connectHandle);
            ERRORMSG("New connectHandle:"<<connectHandle<<endl);
            if(connected != OK)
            {
                DEBUGMSG("Object do_connect failed, rc:"<<connected<<endl);
                ERRORMSG("Object do_connect failed, rc:"<<connected<<endl);
                return connected;
            }
            DEBUGMSG("Object connection complete,
connectHandle:"<<DEBUGADDRESS(connectHandle)<<endl);
            connectHandleInUse = 0;
            return OK;
        }
        else
        {
            DEBUGMSG("Object's connectHandle already in use, connect failed"<<endl);
            ERRORMSG("Object's connectHandle already in use, connect failed"<<endl);
            return ERR_HANDLE_IN_USE;
        }
    }

    DEBUGMSG("Object already has connection established."<<endl);
    return OK;
}

tpccCom/tpccCom_p.c

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 7.00.0500 */
/* at Thu Apr 01 11:20:10 2010
*/
/*
** Compiler settings for .tpccCom.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec( ) decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
@@@MIDL_FILE_HEADING( )

```

```

#if !defined(_M_IA64) && !defined(_M_AMD64)

#pragma warning( disable: 4049 ) /* more than 64k source lines */
#if _MSC_VER >= 1200
#pragma warning(push)
#endif

#pragma warning( disable: 4211 ) /* redefine extern to static */
#pragma warning( disable: 4232 ) /* dllimport identity */
#pragma warning( disable: 4024 ) /* array to pointer mapping */
#pragma warning( disable: 4152 ) /* function/data pointer conversion in expression */
#pragma warning( disable: 4100 ) /* unreferenced arguments in x86 call */

#pragma optimize("", off)

#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this file */
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpccCom.h"

#define TYPE_FORMAT_STRING_SIZE 1089
#define PROC_FORMAT_STRING_SIZE 409
#define EXPR_FORMAT_STRING_SIZE 1
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 2

typedef struct _tpccCom_MIDL_TYPE_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} tpccCom_MIDL_TYPE_FORMAT_STRING;

typedef struct _tpccCom_MIDL_PROC_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} tpccCom_MIDL_PROC_FORMAT_STRING;

typedef struct _tpccCom_MIDL_EXPR_FORMAT_STRING
{
    long Pad;
    unsigned char Format[ EXPR_FORMAT_STRING_SIZE ];
} tpccCom_MIDL_EXPR_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x10,0x48,0x60}},{2,0}};

extern const tpccCom_MIDL_TYPE_FORMAT_STRING tpccCom_MIDL_TypeFormatString;
extern const tpccCom_MIDL_PROC_FORMAT_STRING tpccCom_MIDL_ProcFormatString;
extern const tpccCom_MIDL_EXPR_FORMAT_STRING tpccCom_MIDL_ExprFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO IComponentRegistrar_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO IComponentRegistrar_ProxyInfo;

```

```
extern const MIDL_STUB_DESC Object_StubDesc;
```

```
extern const MIDL_SERVER_INFO Itpcc_com_ServerInfo;  
extern const MIDL_STUBLESS_PROXY_INFO Itpcc_com_ProxyInfo;
```

```
extern const USER_MARSHAL_ROUTINE_QUADRUPLE  
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];
```

```
#if !defined(__RPC_WIN32__)  
#error Invalid build platform for this stub.  
#endif
```

```
#if !(TARGET_IS_NT50_OR_LATER)  
#error You need a Windows 2000 or later to run this stub because it uses these features:  
#error /robust command line switch.  
#error However, your C/C++ compilation flags indicate you intend to run this app on earlier systems.  
#error This app will fail with the RPC_X_WRONG_STUB_VERSION error.  
#endif
```

```
static const tpccCom_MIDL_PROC_FORMAT_STRING tpccCom__MIDL_ProcFormatString =
```

```
{  
  0,  
  {  
    /* Procedure Attach */  
    0x33, /* FC_AUTO_HANDLE */  
    0x6c, /* Old Flags: object, Oi2 */  
    /* 2 */ NdrFcLong( 0x0 ), /* 0 */  
    /* 6 */ NdrFcShort( 0x7 ), /* 7 */  
    /* 8 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */  
    /* 10 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 12 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 14 */ 0x46, /* Oi2 Flags: clt must size, has return, has ext, */  
    0x2, /* 2 */  
    /* 16 */ 0x8, /* 8 */  
    0x5, /* Ext Flags: new corr desc, srv corr check, */  
    /* 18 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 20 */ NdrFcShort( 0x1 ), /* 1 */  
    /* 22 */ NdrFcShort( 0x0 ), /* 0 */  
    /* Parameter bstrPath */  
    /* 24 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */  
    /* 26 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */  
    /* 28 */ NdrFcShort( 0x1c ), /* Type Offset=28 */  
    /* Return value */  
    /* 30 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */  
    /* 32 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */  
    /* 34 */ 0x8, /* FC_LONG */  
    0x0, /* 0 */  
    /* Procedure doSetComplete */  
    /* Procedure RegisterAll */  
    /* 36 */ 0x33, /* FC_AUTO_HANDLE */  
    0x6c, /* Old Flags: object, Oi2 */  
    /* 38 */ NdrFcLong( 0x0 ), /* 0 */  
    /* 42 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 44 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */  
    /* 46 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 48 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 50 */ 0x44, /* Oi2 Flags: has return, has ext, */  
    0x1, /* 1 */
```

```
/* 52 */ 0x8, /* 8 */  
 0x1, /* Ext Flags: new corr desc, */  
    /* 54 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 56 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 58 */ NdrFcShort( 0x0 ), /* 0 */  
    /* Return value */  
    /* Return value */  
    /* 60 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */  
    /* 62 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */  
    /* 64 */ 0x8, /* FC_LONG */  
    0x0, /* 0 */  
    /* Procedure UnregisterAll */  
    /* 66 */ 0x33, /* FC_AUTO_HANDLE */  
    0x6c, /* Old Flags: object, Oi2 */  
    /* 68 */ NdrFcLong( 0x0 ), /* 0 */  
    /* 72 */ NdrFcShort( 0x9 ), /* 9 */  
    /* 74 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */  
    /* 76 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 78 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 80 */ 0x44, /* Oi2 Flags: has return, has ext, */  
    0x1, /* 1 */  
    /* 82 */ 0x8, /* 8 */  
    0x1, /* Ext Flags: new corr desc, */  
    /* 84 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 86 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 88 */ NdrFcShort( 0x0 ), /* 0 */  
    /* Return value */  
    /* 90 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */  
    /* 92 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */  
    /* 94 */ 0x8, /* FC_LONG */  
    0x0, /* 0 */  
    /* Procedure GetComponent */  
    /* 96 */ 0x33, /* FC_AUTO_HANDLE */  
    0x6c, /* Old Flags: object, Oi2 */  
    /* 98 */ NdrFcLong( 0x0 ), /* 0 */  
    /* 102 */ NdrFcShort( 0xa ), /* 10 */  
    /* 104 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */  
    /* 106 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 108 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 110 */ 0x45, /* Oi2 Flags: srv must size, has return, has ext, */  
    0x3, /* 3 */  
    /* 112 */ 0x8, /* 8 */  
    0x3, /* Ext Flags: new corr desc, clt corr check, */  
    /* 114 */ NdrFcShort( 0xb3 ), /* 119 */  
    /* 116 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 118 */ NdrFcShort( 0x0 ), /* 0 */  
    /* Parameter pbstrCLSIDs */  
    /* 120 */ NdrFcShort( 0x2113 ), /* Flags: must size, must free, out, simple ref, srv alloc size=8 */  
    /* 122 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */  
    /* 124 */ NdrFcShort( 0x41e ), /* Type Offset=1054 */  
    /* Parameter pbstrDescriptions */  
    /* 126 */ NdrFcShort( 0x2113 ), /* Flags: must size, must free, out, simple ref, srv alloc size=8 */  
    /* 128 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */  
    /* 130 */ NdrFcShort( 0x41e ), /* Type Offset=1054 */  
    /* Return value */  
    /* 132 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */  
    /* 134 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
```

```
/* 136 */ 0x8, /* FC_LONG */  
 0x0, /* 0 */  
    /* Procedure RegisterComponent */  
    /* 138 */ 0x33, /* FC_AUTO_HANDLE */  
    0x6c, /* Old Flags: object, Oi2 */  
    /* 140 */ NdrFcLong( 0x0 ), /* 0 */  
    /* 144 */ NdrFcShort( 0xb ), /* 11 */  
    /* 146 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */  
    /* 148 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 150 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 152 */ 0x46, /* Oi2 Flags: clt must size, has return, has ext, */  
    0x2, /* 2 */  
    /* 154 */ 0x8, /* 8 */  
    0x5, /* Ext Flags: new corr desc, srv corr check, */  
    /* 156 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 158 */ NdrFcShort( 0x1 ), /* 1 */  
    /* 160 */ NdrFcShort( 0x0 ), /* 0 */  
    /* Parameter bstrCLSID */  
    /* 162 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */  
    /* 164 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */  
    /* 166 */ NdrFcShort( 0x1c ), /* Type Offset=28 */  
    /* Return value */  
    /* 168 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */  
    /* 170 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */  
    /* 172 */ 0x8, /* FC_LONG */  
    0x0, /* 0 */  
    /* Procedure UnregisterComponent */  
    /* 174 */ 0x33, /* FC_AUTO_HANDLE */  
    0x6c, /* Old Flags: object, Oi2 */  
    /* 176 */ NdrFcLong( 0x0 ), /* 0 */  
    /* 180 */ NdrFcShort( 0xc ), /* 12 */  
    /* 182 */ NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */  
    /* 184 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 186 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 188 */ 0x46, /* Oi2 Flags: clt must size, has return, has ext, */  
    0x2, /* 2 */  
    /* 190 */ 0x8, /* 8 */  
    0x5, /* Ext Flags: new corr desc, srv corr check, */  
    /* 192 */ NdrFcShort( 0x0 ), /* 0 */  
    /* 194 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 196 */ NdrFcShort( 0x0 ), /* 0 */  
    /* Parameter bstrCLSID */  
    /* 198 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */  
    /* 200 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */  
    /* 202 */ NdrFcShort( 0x1c ), /* Type Offset=28 */  
    /* Return value */  
    /* 204 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */  
    /* 206 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */  
    /* 208 */ 0x8, /* FC_LONG */  
    0x0, /* 0 */  
    /* Procedure doStockLevel */  
    /* 210 */ 0x33, /* FC_AUTO_HANDLE */  
    0x6c, /* Old Flags: object, Oi2 */  
    /* 212 */ NdrFcLong( 0x0 ), /* 0 */  
    /* 216 */ NdrFcShort( 0x3 ), /* 3 */  
    /* 218 */ NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */  
    /* 220 */ NdrFcShort( 0x1c ), /* 28 */  
    /* 222 */ NdrFcShort( 0x8 ), /* 8 */  
    /* 224 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
```

```

/* 226 */0x8,      0x3,      /* 3 */
/* 228 */NdrFcShort( 0x1 ), /* 1 */
/* 230 */NdrFcShort( 0x1 ), /* 1 */
/* 232 */NdrFcShort( 0x0 ), /* 0 */

/* Parameter size */

/* 234 */NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 236 */NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 238 */0x8, /* FC_LONG */
/* 240 */0x0, /* 0 */

/* Parameter buffer */

/* 240 */NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out, srv alloc size=8 */
/* 242 */NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 244 */NdrFcShort( 0x42c ), /* Type Offset=1068 */

/* Return value */

/* 246 */NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 248 */NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 250 */0x8, /* FC_LONG */
/* 252 */0x0, /* 0 */

/* Procedure doNewOrder */

/* 252 */0x33, /* FC_AUTO_HANDLE */
/* 254 */0x6c, /* Old Flags: object, Oi2 */
/* 254 */NdrFcLong( 0x0 ), /* 0 */
/* 258 */NdrFcShort( 0x4 ), /* 4 */
/* 260 */NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 262 */NdrFcShort( 0x1c ), /* 28 */
/* 264 */NdrFcShort( 0x8 ), /* 8 */
/* 266 */0x47, /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
/* 268 */0x3, /* 3 */
/* 268 */0x8, /* 8 */
/* 270 */0x7, /* Ext Flags: new corr desc, clt corr check, srv corr check, */
/* 270 */NdrFcShort( 0x1 ), /* 1 */
/* 272 */NdrFcShort( 0x1 ), /* 1 */
/* 274 */NdrFcShort( 0x0 ), /* 0 */

/* Parameter size */

/* 276 */NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 278 */NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 280 */0x8, /* FC_LONG */
/* 282 */0x0, /* 0 */

/* Parameter buffer */

/* 282 */NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out, srv alloc size=8 */
/* 284 */NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 286 */NdrFcShort( 0x42c ), /* Type Offset=1068 */

/* Return value */

/* 288 */NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 290 */NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 292 */0x8, /* FC_LONG */
/* 294 */0x0, /* 0 */

/* Procedure doPayment */

/* 294 */0x33, /* FC_AUTO_HANDLE */
/* 296 */0x6c, /* Old Flags: object, Oi2 */
/* 296 */NdrFcLong( 0x0 ), /* 0 */
/* 300 */NdrFcShort( 0x5 ), /* 5 */
/* 302 */NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 304 */NdrFcShort( 0x1c ), /* 28 */
/* 306 */NdrFcShort( 0x8 ), /* 8 */

```

```

/* 308 */0x47, /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
/* 310 */0x8, /* 8 */
/* 312 */NdrFcShort( 0x1 ), /* 1 */
/* 314 */NdrFcShort( 0x1 ), /* 1 */
/* 316 */NdrFcShort( 0x0 ), /* 0 */

/* Parameter size */

/* 318 */NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 320 */NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 322 */0x8, /* FC_LONG */
/* 324 */0x0, /* 0 */

/* Parameter buffer */

/* 324 */NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out, srv alloc size=8 */
/* 326 */NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 328 */NdrFcShort( 0x42c ), /* Type Offset=1068 */

/* Return value */

/* 330 */NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 332 */NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 334 */0x8, /* FC_LONG */
/* 336 */0x0, /* 0 */

/* Procedure doOrderStatus */

/* 336 */0x33, /* FC_AUTO_HANDLE */
/* 338 */0x6c, /* Old Flags: object, Oi2 */
/* 338 */NdrFcLong( 0x0 ), /* 0 */
/* 342 */NdrFcShort( 0x6 ), /* 6 */
/* 344 */NdrFcShort( 0x10 ), /* x86 Stack size/offset = 16 */
/* 346 */NdrFcShort( 0x1c ), /* 28 */
/* 348 */NdrFcShort( 0x8 ), /* 8 */
/* 350 */0x47, /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
/* 352 */0x3, /* 3 */
/* 352 */0x8, /* 8 */
/* 354 */NdrFcShort( 0x1 ), /* 1 */
/* 356 */NdrFcShort( 0x1 ), /* 1 */
/* 358 */NdrFcShort( 0x0 ), /* 0 */

/* Parameter size */

/* 360 */NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 362 */NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 364 */0x8, /* FC_LONG */
/* 366 */0x0, /* 0 */

/* Parameter buffer */

/* 366 */NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out, srv alloc size=8 */
/* 368 */NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 370 */NdrFcShort( 0x42c ), /* Type Offset=1068 */

/* Return value */

/* 372 */NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 374 */NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 376 */0x8, /* FC_LONG */
/* 378 */0x0, /* 0 */

/* Procedure doDBInfo */

/* 378 */0x33, /* FC_AUTO_HANDLE */
/* 380 */0x6c, /* Old Flags: object, Oi2 */
/* 380 */NdrFcLong( 0x0 ), /* 0 */
/* 384 */NdrFcShort( 0x7 ), /* 7 */
/* 386 */NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 388 */NdrFcShort( 0x0 ), /* 0 */

```

```

/* 390 */NdrFcShort( 0x8 ), /* 8 */
/* 392 */0x44, /* Oi2 Flags: has return, has ext, */
/* 394 */0x8, /* 8 */
/* 396 */NdrFcShort( 0x0 ), /* 0 */
/* 398 */NdrFcShort( 0x0 ), /* 0 */
/* 400 */NdrFcShort( 0x0 ), /* 0 */

/* Return value */

/* 402 */NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 404 */NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 406 */0x8, /* FC_LONG */
/* 408 */0x0, /* 0 */

}
};

static const tpcCom_MIDL_TYPE_FORMAT_STRING tpcCom_MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
/* 2 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0xe ), /* Offset= 14 (18) */
/* 6 */
0x1b, /* FC_CARRAY */
0x1, /* 1 */
/* 8 */ NdrFcShort( 0x2 ), /* 2 */
/* 10 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /* */
/* 12 */ NdrFcShort( 0xffff ), /* -4 */
/* 14 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 16 */ 0x6, /* FC_SHORT */
/* 18 */
0x5b, /* FC_END */
0x17, /* FC_CSTRUCT */
0x3, /* 3 */
/* 20 */ NdrFcShort( 0x8 ), /* 8 */
/* 22 */ NdrFcShort( 0xffff0 ), /* Offset= -16 (6) */
/* 24 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 26 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 28 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /* 131 */
/* 30 */ NdrFcShort( 0x0 ), /* 0 */
/* 32 */ NdrFcShort( 0x4 ), /* 4 */
/* 34 */ NdrFcShort( 0x0 ), /* 0 */
/* 36 */ NdrFcShort( 0xffde ), /* Offset= -34 (2) */
/* 38 */
0x11, 0x4, /* FC_RP [allocated_on_stack] */
/* 40 */ NdrFcShort( 0x3f6 ), /* Offset= 1014 (1054) */
/* 42 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 44 */ NdrFcShort( 0x2 ), /* Offset= 2 (46) */
/* 46 */
0x13, 0x0, /* FC_OP */
/* 48 */ NdrFcShort( 0x3dc ), /* Offset= 988 (1036) */
/* 50 */
0x2a, /* FC_ENCAPSULATED_UNION */
0x49, /* 73 */
/* 52 */ NdrFcShort( 0x18 ), /* 24 */
/* 54 */ NdrFcShort( 0xa ), /* 10 */
/* 56 */ NdrFcLong( 0x8 ), /* 8 */
/* 60 */ NdrFcShort( 0x5a ), /* Offset= 90 (150) */
/* 62 */ NdrFcLong( 0xd ), /* 13 */
/* 66 */ NdrFcShort( 0x90 ), /* Offset= 144 (210) */
/* 68 */ NdrFcLong( 0x9 ), /* 9 */
/* 72 */ NdrFcShort( 0xc2 ), /* Offset= 194 (266) */

```

```

/* 74 */ NdrFcLong( 0xc ), /* 12 */
/* 78 */ NdrFcShort( 0x2c0 ), /* Offset= 704 (782) */
/* 80 */ NdrFcLong( 0x24 ), /* 36 */
/* 84 */ NdrFcShort( 0x2ea ), /* Offset= 746 (830) */
/* 86 */ NdrFcLong( 0x800d ), /* 32781 */
/* 90 */ NdrFcShort( 0x306 ), /* Offset= 774 (864) */
/* 92 */ NdrFcLong( 0x10 ), /* 16 */
/* 96 */ NdrFcShort( 0x320 ), /* Offset= 800 (896) */
/* 98 */ NdrFcLong( 0x2 ), /* 2 */
/* 102 */ NdrFcShort( 0x33a ), /* Offset= 826 (928) */
/* 104 */ NdrFcLong( 0x3 ), /* 3 */
/* 108 */ NdrFcShort( 0x354 ), /* Offset= 852 (960) */
/* 110 */ NdrFcLong( 0x14 ), /* 20 */
/* 114 */ NdrFcShort( 0x36e ), /* Offset= 878 (992) */
/* 116 */ NdrFcShort( 0xffff ), /* Offset= -1 (115) */
/* 118 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 120 */ NdrFcShort( 0x4 ), /* 4 */
/* 122 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 124 */ NdrFcShort( 0x0 ), /* 0 */
/* 126 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 128 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 130 */
0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 132 */ NdrFcShort( 0x4 ), /* 4 */
/* 134 */ NdrFcShort( 0x0 ), /* 0 */
/* 136 */ NdrFcShort( 0x1 ), /* 1 */
/* 138 */ NdrFcShort( 0x0 ), /* 0 */
/* 140 */ NdrFcShort( 0x0 ), /* 0 */
/* 142 */ 0x13, 0x0, /* FC_OP */
/* 144 */ NdrFcShort( 0xffff2 ), /* Offset= -126 (18) */
/* 146 */
0x5b, /* FC_END */
/* 148 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 150 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 152 */ NdrFcShort( 0x8 ), /* 8 */
/* 154 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 156 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 158 */ NdrFcShort( 0x4 ), /* 4 */
/* 160 */ NdrFcShort( 0x4 ), /* 4 */
/* 162 */ 0x11, 0x0, /* FC_RP */
/* 164 */ NdrFcShort( 0xf1d2 ), /* Offset= -46 (118) */
/* 166 */
0x5b, /* FC_END */
/* 168 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 170 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x0 ), /* 0 */
/* 178 */ NdrFcShort( 0x0 ), /* 0 */
/* 180 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 182 */ 0x0, /* 0 */
0x0, /* 0 */
/* 184 */ 0x0, /* 0 */

```

```

0x0, /* 0 */
/* 186 */ 0x0, /* 0 */
0x46, /* 70 */
/* 188 */
0x21, /* FC_BOGUS_ARRAY */
0x3, /* 3 */
/* 190 */ NdrFcShort( 0x0 ), /* 0 */
/* 192 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 194 */ NdrFcShort( 0x0 ), /* 0 */
/* 196 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 198 */ NdrFcLong( 0xfffff ), /* -1 */
/* 202 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 204 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 206 */ NdrFcShort( 0xffdc ), /* Offset= -36 (170) */
/* 208 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 210 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 212 */ NdrFcShort( 0x8 ), /* 8 */
/* 214 */ NdrFcShort( 0x0 ), /* 0 */
/* 216 */ NdrFcShort( 0x6 ), /* Offset= 6 (222) */
/* 218 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 220 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 222 */
0x11, 0x0, /* FC_RP */
/* 224 */ NdrFcShort( 0xffdc ), /* Offset= -36 (188) */
/* 226 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 228 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 232 */ NdrFcShort( 0x0 ), /* 0 */
/* 234 */ NdrFcShort( 0x0 ), /* 0 */
/* 236 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 238 */ 0x0, /* 0 */
0x0, /* 0 */
/* 240 */ 0x0, /* 0 */
0x0, /* 0 */
/* 242 */ 0x0, /* 0 */
0x46, /* 70 */
/* 244 */
0x21, /* FC_BOGUS_ARRAY */
0x3, /* 3 */
/* 246 */ NdrFcShort( 0x0 ), /* 0 */
/* 248 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 250 */ NdrFcShort( 0x0 ), /* 0 */
/* 252 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 254 */ NdrFcLong( 0xfffff ), /* -1 */
/* 258 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 260 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 262 */ NdrFcShort( 0xffdc ), /* Offset= -36 (226) */
/* 264 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 266 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 268 */ NdrFcShort( 0x8 ), /* 8 */
/* 270 */ NdrFcShort( 0x0 ), /* 0 */
/* 272 */ NdrFcShort( 0x6 ), /* Offset= 6 (278) */
/* 274 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 276 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 278 */
0x11, 0x0, /* FC_RP */
/* 280 */ NdrFcShort( 0xffdc ), /* Offset= -36 (244) */

```

```

/* 282 */
0x2b, /* FC_NON_ENCAPSULATED_UNION */
0x9, /* FC_ULONG */
/* 284 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
/* 286 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 288 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 290 */ NdrFcShort( 0x2 ), /* Offset= 2 (292) */
/* 292 */ NdrFcShort( 0x10 ), /* 16 */
/* 294 */ NdrFcShort( 0x2f ), /* 47 */
/* 296 */ NdrFcLong( 0x14 ), /* 20 */
/* 300 */ NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 302 */ NdrFcLong( 0x3 ), /* 3 */
/* 306 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 308 */ NdrFcLong( 0x11 ), /* 17 */
/* 312 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 314 */ NdrFcLong( 0x2 ), /* 2 */
/* 318 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 320 */ NdrFcLong( 0x4 ), /* 4 */
/* 324 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 326 */ NdrFcLong( 0x5 ), /* 5 */
/* 330 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 332 */ NdrFcLong( 0xb ), /* 11 */
/* 336 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 338 */ NdrFcLong( 0xa ), /* 10 */
/* 342 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 344 */ NdrFcLong( 0x6 ), /* 6 */
/* 348 */ NdrFcShort( 0xe8 ), /* Offset= 232 (580) */
/* 350 */ NdrFcLong( 0x7 ), /* 7 */
/* 354 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 356 */ NdrFcLong( 0x8 ), /* 8 */
/* 360 */ NdrFcShort( 0xe2 ), /* Offset= 226 (586) */
/* 362 */ NdrFcLong( 0xd ), /* 13 */
/* 366 */ NdrFcShort( 0xffff3c ), /* Offset= -196 (170) */
/* 368 */ NdrFcLong( 0x9 ), /* 9 */
/* 372 */ NdrFcShort( 0xffff6e ), /* Offset= -146 (226) */
/* 374 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 378 */ NdrFcShort( 0xd4 ), /* Offset= 212 (590) */
/* 380 */ NdrFcLong( 0x24 ), /* 36 */
/* 384 */ NdrFcShort( 0xd6 ), /* Offset= 214 (598) */
/* 386 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 390 */ NdrFcShort( 0xd0 ), /* Offset= 208 (598) */
/* 392 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 396 */ NdrFcShort( 0x100 ), /* Offset= 256 (652) */
/* 398 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 402 */ NdrFcShort( 0xfe ), /* Offset= 254 (656) */
/* 404 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 408 */ NdrFcShort( 0xfc ), /* Offset= 252 (660) */
/* 410 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 414 */ NdrFcShort( 0xfa ), /* Offset= 250 (664) */
/* 416 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 420 */ NdrFcShort( 0xf8 ), /* Offset= 248 (668) */
/* 422 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 426 */ NdrFcShort( 0xf6 ), /* Offset= 246 (672) */
/* 428 */ NdrFcLong( 0x400b ), /* 16395 */
/* 432 */ NdrFcShort( 0xe0 ), /* Offset= 224 (656) */
/* 434 */ NdrFcLong( 0x400a ), /* 16394 */
/* 438 */ NdrFcShort( 0xde ), /* Offset= 222 (660) */
/* 440 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 444 */ NdrFcShort( 0xe8 ), /* Offset= 232 (676) */
/* 446 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 450 */ NdrFcShort( 0xde ), /* Offset= 222 (672) */
/* 452 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 456 */ NdrFcShort( 0xe0 ), /* Offset= 224 (680) */
/* 458 */ NdrFcLong( 0x400d ), /* 16397 */
/* 462 */ NdrFcShort( 0xde ), /* Offset= 222 (684) */
/* 464 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 468 */ NdrFcShort( 0xdc ), /* Offset= 220 (688) */
/* 470 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 474 */ NdrFcShort( 0xda ), /* Offset= 218 (692) */
/* 476 */ NdrFcLong( 0x400c ), /* 16396 */
/* 480 */ NdrFcShort( 0xe0 ), /* Offset= 224 (704) */
/* 482 */ NdrFcLong( 0x10 ), /* 16 */

```



```

/* 486 */NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 488 */NdrFcLong( 0x12 ), /* 18 */
/* 492 */NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 494 */NdrFcLong( 0x13 ), /* 19 */
/* 498 */NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 500 */NdrFcLong( 0x15 ), /* 21 */
/* 504 */NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 506 */NdrFcLong( 0x16 ), /* 22 */
/* 510 */NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 512 */NdrFcLong( 0x17 ), /* 23 */
/* 516 */NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 518 */NdrFcLong( 0xe ), /* 14 */
/* 522 */NdrFcShort( 0xbe ), /* Offset= 190 (712) */
/* 524 */NdrFcLong( 0x400e ), /* 16398 */
/* 528 */NdrFcShort( 0xc2 ), /* Offset= 194 (722) */
/* 530 */NdrFcLong( 0x4010 ), /* 16400 */
/* 534 */NdrFcShort( 0xc0 ), /* Offset= 192 (726) */
/* 536 */NdrFcLong( 0x4012 ), /* 16402 */
/* 540 */NdrFcShort( 0x74 ), /* Offset= 116 (656) */
/* 542 */NdrFcLong( 0x4013 ), /* 16403 */
/* 546 */NdrFcShort( 0x72 ), /* Offset= 114 (660) */
/* 548 */NdrFcLong( 0x4015 ), /* 16405 */
/* 552 */NdrFcShort( 0x70 ), /* Offset= 112 (664) */
/* 554 */NdrFcLong( 0x4016 ), /* 16406 */
/* 558 */NdrFcShort( 0x66 ), /* Offset= 102 (660) */
/* 560 */NdrFcLong( 0x4017 ), /* 16407 */
/* 564 */NdrFcShort( 0x60 ), /* Offset= 96 (660) */
/* 566 */NdrFcLong( 0x0 ), /* 0 */
/* 570 */NdrFcShort( 0x0 ), /* Offset= 0 (570) */
/* 572 */NdrFcLong( 0x1 ), /* 1 */
/* 576 */NdrFcShort( 0x0 ), /* Offset= 0 (576) */
/* 578 */NdrFcShort( 0xffff ), /* Offset= -1 (577) */
/* 580 */
0x15, /* FC_STRUCT */
0x7, /* 7 */
/* 582 */NdrFcShort( 0x8 ), /* 8 */
/* 584 */0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 586 */
0x13, 0x0, /* FC_OP */
/* 588 */NdrFcShort( 0xfdc6 ), /* Offset= -570 (18) */
/* 590 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 592 */NdrFcShort( 0x2 ), /* Offset= 2 (594) */
/* 594 */
0x13, 0x0, /* FC_OP */
/* 596 */NdrFcShort( 0x1b8 ), /* Offset= 440 (1036) */
/* 598 */
0x13, 0x0, /* FC_OP */
/* 600 */NdrFcShort( 0x20 ), /* Offset= 32 (632) */
/* 602 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 604 */NdrFcLong( 0x2f ), /* 47 */
/* 608 */NdrFcShort( 0x0 ), /* 0 */
/* 610 */NdrFcShort( 0x0 ), /* 0 */
/* 612 */0xc0, /* 192 */
0x0, /* 0 */
/* 614 */0x0, /* 0 */
0x0, /* 0 */
/* 616 */0x0, /* 0 */
0x0, /* 0 */
/* 618 */0x0, /* 0 */
0x46, /* 70 */
/* 620 */
0x1b, /* FC_CARRAY */
0x0, /* 0 */
/* 622 */NdrFcShort( 0x1 ), /* 1 */
/* 624 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* 0 */
/* 626 */NdrFcShort( 0x4 ), /* 4 */
/* 628 */NdrFcShort( 0x1 ), /* Corr flags: early */
/* 630 */0x1, /* FC_BYTE */

```

```

0x5b, /* FC_END */
/* 632 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 634 */NdrFcShort( 0x10 ), /* 16 */
/* 636 */NdrFcShort( 0x0 ), /* 0 */
/* 638 */NdrFcShort( 0xa ), /* Offset= 10 (648) */
/* 640 */0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 642 */0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 644 */NdrFcShort( 0xffd6 ), /* Offset= -42 (602) */
/* 646 */0x36, /* FC_POINTER */
0x5b, /* FC_END */
/* 648 */
0x13, 0x0, /* FC_OP */
/* 650 */NdrFcShort( 0xffe2 ), /* Offset= -30 (620) */
/* 652 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 654 */0x1, /* FC_BYTE */
0x5c, /* FC_PAD */
/* 656 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 658 */0x6, /* FC_SHORT */
0x5c, /* FC_PAD */
/* 660 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 662 */0x8, /* FC_LONG */
0x5c, /* FC_PAD */
/* 664 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 666 */0xb, /* FC_HYPER */
0x5c, /* FC_PAD */
/* 668 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 670 */0xa, /* FC_FLOAT */
0x5c, /* FC_PAD */
/* 672 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 674 */0xc, /* FC_DOUBLE */
0x5c, /* FC_PAD */
/* 676 */
0x13, 0x0, /* FC_OP */
/* 678 */NdrFcShort( 0xff9e ), /* Offset= -98 (580) */
/* 680 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 682 */NdrFcShort( 0xffa0 ), /* Offset= -96 (586) */
/* 684 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 686 */NdrFcShort( 0xfdfc ), /* Offset= -516 (170) */
/* 688 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 690 */NdrFcShort( 0xfef30 ), /* Offset= -464 (226) */
/* 692 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 694 */NdrFcShort( 0x2 ), /* Offset= 2 (696) */
/* 696 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 698 */NdrFcShort( 0x2 ), /* Offset= 2 (700) */
/* 700 */
0x13, 0x0, /* FC_OP */
/* 702 */NdrFcShort( 0x14e ), /* Offset= 334 (1036) */
/* 704 */
0x13, 0x10, /* FC_OP [pointer_deref] */
/* 706 */NdrFcShort( 0x2 ), /* Offset= 2 (708) */
/* 708 */
0x13, 0x0, /* FC_OP */
/* 710 */NdrFcShort( 0x14 ), /* Offset= 20 (730) */
/* 712 */
0x15, /* FC_STRUCT */
0x7, /* 7 */
/* 714 */NdrFcShort( 0x10 ), /* 16 */
/* 716 */0x6, /* FC_SHORT */

```

```

0x1, /* FC_BYTE */
/* 718 */0x1, /* FC_BYTE */
0x8, /* FC_LONG */
/* 720 */0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 722 */
0x13, 0x0, /* FC_OP */
/* 724 */NdrFcShort( 0xffff4 ), /* Offset= -12 (712) */
/* 726 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 728 */0x2, /* FC_CHAR */
0x5c, /* FC_PAD */
/* 730 */
0x1a, /* FC_BOGUS_STRUCT */
0x7, /* 7 */
/* 732 */NdrFcShort( 0x20 ), /* 32 */
/* 734 */NdrFcShort( 0x0 ), /* 0 */
/* 736 */NdrFcShort( 0x0 ), /* Offset= 0 (736) */
/* 738 */0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 740 */0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 742 */0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 744 */0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 746 */NdrFcShort( 0xfef30 ), /* Offset= -464 (282) */
/* 748 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 750 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 752 */NdrFcShort( 0x4 ), /* 4 */
/* 754 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* 0 */
/* 756 */NdrFcShort( 0x0 ), /* 0 */
/* 758 */NdrFcShort( 0x1 ), /* Corr flags: early */
/* 760 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 762 */
0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 764 */NdrFcShort( 0x4 ), /* 4 */
/* 766 */NdrFcShort( 0x0 ), /* 0 */
/* 768 */NdrFcShort( 0x1 ), /* 1 */
/* 770 */NdrFcShort( 0x0 ), /* 0 */
/* 772 */NdrFcShort( 0x0 ), /* 0 */
/* 774 */0x13, 0x0, /* FC_OP */
/* 776 */NdrFcShort( 0xffd2 ), /* Offset= -46 (730) */
/* 778 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 780 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 782 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 784 */NdrFcShort( 0x8 ), /* 8 */
/* 786 */NdrFcShort( 0x0 ), /* 0 */
/* 788 */NdrFcShort( 0x6 ), /* Offset= 6 (794) */
/* 790 */0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 792 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 794 */
0x11, 0x0, /* FC_RP */
/* 796 */NdrFcShort( 0xffd2 ), /* Offset= -46 (750) */
/* 798 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 800 */NdrFcShort( 0x4 ), /* 4 */

```

```

/* 802 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 804 */NdrFcShort( 0x0 ), /* 0 */
/* 806 */NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 808 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 810 */
0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 812 */NdrFcShort( 0x4 ), /* 4 */
/* 814 */NdrFcShort( 0x0 ), /* 0 */
/* 816 */NdrFcShort( 0x1 ), /* 1 */
/* 818 */NdrFcShort( 0x0 ), /* 0 */
/* 820 */NdrFcShort( 0x0 ), /* 0 */
/* 822 */0x13, 0x0, /* FC_OP */
/* 824 */NdrFcShort( 0xff40 ), /* Offset= -192 (632) */
/* 826 */
0x5b, /* FC_END */
/* 828 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 830 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 832 */NdrFcShort( 0x8 ), /* 8 */
/* 834 */NdrFcShort( 0x0 ), /* 0 */
/* 836 */NdrFcShort( 0x6 ), /* Offset= 6 (842) */
/* 838 */0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 840 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 842 */
0x11, 0x0, /* FC_LP */
/* 844 */NdrFcShort( 0xffd2 ), /* Offset= -46 (798) */
/* 846 */
0x1d, /* FC_SMFARRAY */
0x0, /* 0 */
/* 848 */NdrFcShort( 0x8 ), /* 8 */
/* 850 */0x1, /* FC_BYTE */
0x5b, /* FC_END */
/* 852 */
0x15, /* FC_STRUCT */
0x3, /* 3 */
/* 854 */NdrFcShort( 0x10 ), /* 16 */
/* 856 */0x8, /* FC_LONG */
0x6, /* FC_SHORT */
/* 858 */0x6, /* FC_SHORT */
0x4c, /* FC_EMBEDDED_COMPLEX */
/* 860 */0x0, /* 0 */
NdrFcShort( 0xff1 ), /* Offset= -15 (846) */
0x5b, /* FC_END */
/* 864 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 866 */NdrFcShort( 0x18 ), /* 24 */
/* 868 */NdrFcShort( 0x0 ), /* 0 */
/* 870 */NdrFcShort( 0xa ), /* Offset= 10 (880) */
/* 872 */0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 874 */0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 876 */NdrFcShort( 0xffe8 ), /* Offset= -24 (852) */
/* 878 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 880 */
0x11, 0x0, /* FC_LP */
/* 882 */NdrFcShort( 0xf04a ), /* Offset= -694 (188) */
/* 884 */
0x1b, /* FC_CARRY */
0x0, /* 0 */
/* 886 */NdrFcShort( 0x1 ), /* 1 */

```

```

/* 888 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 890 */NdrFcShort( 0x0 ), /* 0 */
/* 892 */NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 894 */0x1, /* FC_BYTE */
0x5b, /* FC_END */
/* 896 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 898 */NdrFcShort( 0x8 ), /* 8 */
/* 900 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 902 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 904 */NdrFcShort( 0x4 ), /* 4 */
/* 906 */NdrFcShort( 0x4 ), /* 4 */
/* 908 */0x13, 0x0, /* FC_OP */
/* 910 */NdrFcShort( 0xffe6 ), /* Offset= -26 (884) */
/* 912 */
0x5b, /* FC_END */
/* 914 */0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 916 */
0x1b, /* FC_CARRY */
0x1, /* 1 */
/* 918 */NdrFcShort( 0x2 ), /* 2 */
/* 920 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 922 */NdrFcShort( 0x0 ), /* 0 */
/* 924 */NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 926 */0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 928 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 930 */NdrFcShort( 0x8 ), /* 8 */
/* 932 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 934 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 936 */NdrFcShort( 0x4 ), /* 4 */
/* 938 */NdrFcShort( 0x4 ), /* 4 */
/* 940 */0x13, 0x0, /* FC_OP */
/* 942 */NdrFcShort( 0xffe6 ), /* Offset= -26 (916) */
/* 944 */
0x5b, /* FC_END */
/* 946 */0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 948 */
0x1b, /* FC_CARRY */
0x3, /* 3 */
/* 950 */NdrFcShort( 0x4 ), /* 4 */
/* 952 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 954 */NdrFcShort( 0x0 ), /* 0 */
/* 956 */NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 958 */0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 960 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 962 */NdrFcShort( 0x8 ), /* 8 */
/* 964 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */

```

```

/* 966 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 968 */NdrFcShort( 0x4 ), /* 4 */
/* 970 */NdrFcShort( 0x4 ), /* 4 */
/* 972 */0x13, 0x0, /* FC_OP */
/* 974 */NdrFcShort( 0xffe6 ), /* Offset= -26 (948) */
/* 976 */
0x5b, /* FC_END */
/* 978 */0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 980 */
0x1b, /* FC_CARRY */
0x7, /* 7 */
/* 982 */NdrFcShort( 0x8 ), /* 8 */
/* 984 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 986 */NdrFcShort( 0x0 ), /* 0 */
/* 988 */NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 990 */0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 992 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 994 */NdrFcShort( 0x8 ), /* 8 */
/* 996 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 998 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 1000 */ NdrFcShort( 0x4 ), /* 4 */
/* 1002 */ NdrFcShort( 0x4 ), /* 4 */
/* 1004 */ 0x13, 0x0, /* FC_OP */
/* 1006 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (980) */
/* 1008 */
0x5b, /* FC_END */
/* 1010 */
0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 1012 */
0x15, /* FC_STRUCT */
0x3, /* 3 */
/* 1014 */ NdrFcShort( 0x8 ), /* 8 */
/* 1016 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 1018 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 1020 */
0x1b, /* FC_CARRY */
0x3, /* 3 */
/* 1022 */ NdrFcShort( 0x8 ), /* 8 */
/* 1024 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
/* 1026 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 1028 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 1030 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 1032 */ NdrFcShort( 0xffec ), /* Offset= -20 (1012) */
/* 1034 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 1036 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 1038 */ NdrFcShort( 0x28 ), /* 40 */
/* 1040 */ NdrFcShort( 0xffec ), /* Offset= -20 (1020) */
/* 1042 */ NdrFcShort( 0x0 ), /* Offset= 0 (1042) */
/* 1044 */ 0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 1046 */ 0x8, /* FC_LONG */

```

```

    0x8, /* FC_LONG */
/* 1048 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
    0x0, /* 0 */
/* 1050 */ NdrFcShort( 0xfc18 ), /* Offset= -1000 (50) */
/* 1052 */ 0x5c, /* FC_PAD */
    0x5b, /* FC_END */
/* 1054 */ 0xb4, /* FC_USER_MARSHAL */
    0x83, /* 131 */
/* 1056 */ NdrFcShort( 0x1 ), /* 1 */
/* 1058 */ NdrFcShort( 0x4 ), /* 4 */
/* 1060 */ NdrFcShort( 0x0 ), /* 0 */
/* 1062 */ NdrFcShort( 0xfc04 ), /* Offset= -1020 (42) */
/* 1064 */
    0x11, 0x8, /* FC_RP [simple_pointer] */
/* 1066 */ 0x8, /* FC_LONG */
    0x5c, /* FC_PAD */
/* 1068 */
    0x11, 0x14, /* FC_RP [allocated_on_stack] [pointer_deref] */
/* 1070 */ NdrFcShort( 0x2 ), /* Offset= 2 (1072) */
/* 1072 */
    0x13, 0x0, /* FC_OP */
/* 1074 */ NdrFcShort( 0x2 ), /* Offset= 2 (1076) */
/* 1076 */
    0x1b, /* FC_CARRAY */
    0x0, /* 0 */
/* 1078 */ NdrFcShort( 0x1 ), /* 1 */
/* 1080 */ 0x28, /* Corr desc: parameter, FC_LONG */
    0x54, /* FC_DEREFERENCE */
/* 1082 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 1084 */ NdrFcShort( 0x1 ), /* Corr flags: early */
/* 1086 */ 0x2, /* FC_CHAR */
    0x5b, /* FC_END */
    0x0
}
};

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        BSTR_UserSize
        ,BSTR_UserMarshal
        ,BSTR_UserUnmarshal
        ,BSTR_UserFree
        },
    {
        LPSAFEARRAY_UserSize
        ,LPSAFEARRAY_UserMarshal
        ,LPSAFEARRAY_UserUnmarshal
        ,LPSAFEARRAY_UserFree
        }
    };

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: IDispatch, ver. 0.0,
GUID={0x00020400,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: IComponentRegistrar, ver. 0.0,
GUID={0xa817e7a2,0x43fa,0x11d0,{0x9e,0x44,0x00,0xaa,0x00,0xb6,0x77,0x0a}} */

#pragma code_seg("orpc")
static const unsigned short IComponentRegistrar_FormatStringOffsetTable[] =
{
    (unsigned short) -1,

```

```

    (unsigned short) -1,
    (unsigned short) -1,
    (unsigned short) -1,
    0,
    36,
    66,
    96,
    138,
    174
};

static const MIDL_STUBLESS_PROXY_INFO IComponentRegistrar_ProxyInfo =
{
    &Object_StubDesc,
    tpccCom__MIDL_ProcFormatString.Format,
    &IComponentRegistrar_FormatStringOffsetTable[-3],
    0,
    0,
    0;
};

static const MIDL_SERVER_INFO IComponentRegistrar_ServerInfo =
{
    &Object_StubDesc,
    0,
    tpccCom__MIDL_ProcFormatString.Format,
    &IComponentRegistrar_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0;
};

CINTERFACE_PROXY_VTABLE(13) _IComponentRegistrarProxyVtbl =
{
    &IComponentRegistrar_ProxyInfo,
    &IID_IComponentRegistrar,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy,
    0 /* (void *) (INT_PTR) -1 */ IDispatch::GetTypeInfoCount */,
    0 /* (void *) (INT_PTR) -1 */ IDispatch::GetTypeInfo */,
    0 /* (void *) (INT_PTR) -1 */ IDispatch::GetIDsOfNames */,
    0 /* IDispatch_Invoke_Proxy */,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::Attach */,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::RegisterAll */,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::UnregisterAll */,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::GetComponents */,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::RegisterComponent */,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::UnregisterComponent */
};

static const PRPC_STUB_FUNCTION IComponentRegistrar_table[] =
{
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2
};

CInterfaceStubVtbl _IComponentRegistrarStubVtbl =
{
    &IID_IComponentRegistrar,
    &IComponentRegistrar_ServerInfo,
    13,
    &IComponentRegistrar_table[-3],
    CStdStubBuffer_DELEGATING_METHODS
};

```

```

};

/* Object interface: Itpcc_com, ver. 0.0,
GUID={0x5B4FA473,0x2E68,0x4D79,{0xA6,0x26,0xF3,0x8B,0x30,0xB8,0x19,0x6E}} */

#pragma code_seg("orpc")
static const unsigned short Itpcc_com_FormatStringOffsetTable[] =
{
    210,
    252,
    294,
    336,
    378,
    36
};

static const MIDL_STUBLESS_PROXY_INFO Itpcc_com_ProxyInfo =
{
    &Object_StubDesc,
    tpccCom__MIDL_ProcFormatString.Format,
    &Itpcc_com_FormatStringOffsetTable[-3],
    0,
    0,
    0;
};

static const MIDL_SERVER_INFO Itpcc_com_ServerInfo =
{
    &Object_StubDesc,
    0,
    tpccCom__MIDL_ProcFormatString.Format,
    &Itpcc_com_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0;
};

CINTERFACE_PROXY_VTABLE(9) _Itpcc_comProxyVtbl =
{
    &Itpcc_com_ProxyInfo,
    &IID_Itpcc_com,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy,
    (void *) (INT_PTR) -1 /* Itpcc_com::doStockLevel */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doNewOrder */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doPayment */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doOrderStatus */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doDBInfo */,
    (void *) (INT_PTR) -1 /* Itpcc_com::doSetComplete */
};

const CInterfaceStubVtbl _Itpcc_comStubVtbl =
{
    &IID_Itpcc_com,
    &Itpcc_com_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    tpccCom__MIDL_TypeFormatString.Format,
};

```

```

1, /* -error_bounds_check flag */
0x50002, /* Ndr library version */
0,
0x70001f4, /* MIDL Version 7.0.500 */
0,
UserMarshalRoutines,
0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* cs routines */
0, /* proxy/server info */
0
};

const CInterfaceProxyVtbl * _tpccCom_ProxyVtblList[] =
{
    (CInterfaceProxyVtbl *) & _tpccCom_ProxyVtbl,
    (CInterfaceProxyVtbl *) & _IComponentRegistrarProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpccCom_StubVtblList[] =
{
    (CInterfaceStubVtbl *) & _tpccCom_StubVtbl,
    (CInterfaceStubVtbl *) & _IComponentRegistrarStubVtbl,
    0
};

PCInterfaceName const _tpccCom_InterfaceNamesList[] =
{
    "tpcc_com",
    "IComponentRegistrar",
    0
};

const IID * _tpccCom_BaselIDList[] =
{
    0,
    &IID_IDispatch,
    0
};

#define _tpccCom_CHECK_IID(n)    IID_GENERIC_CHECK_IID( _tpccCom, pIID, n)

int __stdcall _tpccCom_IID_Lookup( const IID * pIID, int * pIndex )
{
    IID_BS_LOOKUP_SETUP

    IID_BS_LOOKUP_INITIAL_TEST( _tpccCom, 2, 1 )
    IID_BS_LOOKUP_RETURN_RESULT( _tpccCom, 2, *pIndex )
}

const ExtendedProxyFileInfo tpccCom_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpccCom_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpccCom_StubVtblList,
    (const PCInterfaceName *) & _tpccCom_InterfaceNamesList,
    (const IID **) & _tpccCom_BaselIDList,
    & _tpccCom_IID_Lookup,
    2,
    2,
    0, /* table of [async_uid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};
#pragma optimize("", on)
#if _MSC_VER >= 1200
#pragma warning(pop)
#endif

```

```

#endif /* !defined(_M_IA64) && !defined(_M_AMD64) */

tpccCom/tpccCom_i.c

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 7.00.0500 */
/* at Thu Apr 01 11:20:10 2010
*/
/* Compiler settings for .tpccCom.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#pragma warning( disable: 4049 ) /* more than 64k source lines */

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else /* !_MIDL_USE_GUIDDEF_

#ifdef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif /* __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif /* CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

```

```

#endif /* _MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_IComponentRegistrar,0xa817e7a2,0x43fa,0x11d0,0x9e,0x44,0x00,0xaa,0x00,0xb6,0x77,0x0a);

MIDL_DEFINE_GUID(IID,
IID_tpcc_com,0x5b4fa473,0x2e68,0x4d79,0xa6,0x26,0xf3,0x8b,0x30,0xb8,0x19,0x6e);

MIDL_DEFINE_GUID(IID,
LIBID_tpccComLib,0x91f1b8b0,0x89e9,0x457b,0xa2,0x28,0x3e,0x2d,0x6c,0xe3,0xe7,0x52);

MIDL_DEFINE_GUID(CLSID,
CLSID_CompReg,0x90eadaff,0xf8d3,0x4711,0x99,0xa9,0x8a,0xc3,0xc0,0xfe,0x5d,0xb9);

MIDL_DEFINE_GUID(CLSID,
CLSID_tpcc_com,0x5f752bf2,0xf739,0x43d4,0x84,0x92,0x44,0xc1,0x95,0x81,0xc0,0xa1);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

tpccCom/tpccCom.h

/* this ALWAYS GENERATED file contains the definitions for the interfaces */

/* File created by MIDL compiler version 7.00.0500 */
/* at Thu Apr 01 11:20:10 2010
*/
/* Compiler settings for .tpccCom.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifdef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 475
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifdef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif /* __RPCNDR_H_VERSION__

#ifdef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /* COM_NO_WINDOWS_H

```

```

#ifndef __tpccCom_h__
#define __tpccCom_h__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef __IComponentRegistrar_FWD_DEFINED__
#define __IComponentRegistrar_FWD_DEFINED__
typedef interface IComponentRegistrar IComponentRegistrar;
#endif /* __IComponentRegistrar_FWD_DEFINED__ */

#ifndef __tpcc_com_FWD_DEFINED__
#define __tpcc_com_FWD_DEFINED__
typedef interface Itpcc_com Itpcc_com;
#endif /* __tpcc_com_FWD_DEFINED__ */

#ifndef __CompReg_FWD_DEFINED__
#define __CompReg_FWD_DEFINED__

#ifndef __cplusplus
typedef class CompReg CompReg;
#else
typedef struct CompReg CompReg;
#endif /* __cplusplus */

#endif /* __CompReg_FWD_DEFINED__ */

#ifndef __tpcc_com_FWD_DEFINED__
#define __tpcc_com_FWD_DEFINED__

#ifndef __cplusplus
typedef class tpcc_com tpcc_com;
#else
typedef struct tpcc_com tpcc_com;
#endif /* __cplusplus */

#endif /* __tpcc_com_FWD_DEFINED__ */

/* header files for imported files */
#include "oid.h"
#include "ocid.h"

#ifndef __cplusplus
extern "C" {
#endif

#ifndef __IComponentRegistrar_INTERFACE_DEFINED__
#define __IComponentRegistrar_INTERFACE_DEFINED__

/* interface IComponentRegistrar */
/* [unique][helpstring][dual][uuid][object] */

EXTERN_C const IID IID_IComponentRegistrar;

#if defined(__cplusplus) && !defined(CINTERFACE)

MIDL_INTERFACE("a817e7a2-43fa-11d0-9e44-00aa00b6770a")
IComponentRegistrar : public IDispatch
{
public:
    virtual /* [id] */ HRESULT STDMETHODCALLTYPE Attach(
        /* [in] */ BSTR bstrPath) = 0;

    virtual /* [id] */ HRESULT STDMETHODCALLTYPE RegisterAll( void) = 0;

```

```

    virtual /* [id] */ HRESULT STDMETHODCALLTYPE UnregisterAll( void) = 0;

    virtual /* [id] */ HRESULT STDMETHODCALLTYPE GetComponents(
        /* [out] */ SAFEARRAY * *pbastrCLSIDs,
        /* [out] */ SAFEARRAY * *pbstrDescriptions) = 0;

    virtual /* [id] */ HRESULT STDMETHODCALLTYPE RegisterComponent(
        /* [in] */ BSTR bstrCLSID) = 0;

    virtual /* [id] */ HRESULT STDMETHODCALLTYPE UnregisterComponent(
        /* [in] */ BSTR bstrCLSID) = 0;

};

#else /* C style interface */

typedef struct IComponentRegistrarVtbl
{
    BEGIN_INTERFACE

    HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
        IComponentRegistrar * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */
        __RPC__deref_out void **ppvObject);

    ULONG ( STDMETHODCALLTYPE *AddRef )(
        IComponentRegistrar * This);

    ULONG ( STDMETHODCALLTYPE *Release )(
        IComponentRegistrar * This);

    HRESULT ( STDMETHODCALLTYPE *GetTypeInfoCount )(
        IComponentRegistrar * This,
        /* [out] */ UINT *pctinfo);

    HRESULT ( STDMETHODCALLTYPE *GetTypeInfo )(
        IComponentRegistrar * This,
        /* [in] */ UINT iTInfo,
        /* [in] */ LCID lcid,
        /* [out] */ ITypeInfo **ppTInfo);

    HRESULT ( STDMETHODCALLTYPE *GetIDsOfNames )(
        IComponentRegistrar * This,
        /* [in] */ REFIID riid,
        /* [size_is][in] */ LPOLESTR *rgszNames,
        /* [range][in] */ UINT cNames,
        /* [in] */ LCID lcid,
        /* [size_is][out] */ DISPID *rgDispId);

    /* [local] */ HRESULT ( STDMETHODCALLTYPE *Invoke )(
        IComponentRegistrar * This,
        /* [in] */ DISPID dispIdMember,
        /* [in] */ REFIID riid,
        /* [in] */ LCID lcid,
        /* [in] */ WORD wFlags,
        /* [out][in] */ DISPPARAMS *pDispParams,
        /* [out] */ VARIANT *pVarResult,
        /* [out] */ EXCEPINFO *pExcepInfo,
        /* [out] */ UINT *puArgErr);

    /* [id] */ HRESULT ( STDMETHODCALLTYPE *Attach )(
        IComponentRegistrar * This,
        /* [in] */ BSTR bstrPath);

    /* [id] */ HRESULT ( STDMETHODCALLTYPE *RegisterAll )(
        IComponentRegistrar * This);

    /* [id] */ HRESULT ( STDMETHODCALLTYPE *UnregisterAll )(
        IComponentRegistrar * This);

    /* [id] */ HRESULT ( STDMETHODCALLTYPE *GetComponents )(

```

```

        IComponentRegistrar * This,
        /* [out] */ SAFEARRAY * *pbastrCLSIDs,
        /* [out] */ SAFEARRAY * *pbstrDescriptions);

    /* [id] */ HRESULT ( STDMETHODCALLTYPE *RegisterComponent )(
        IComponentRegistrar * This,
        /* [in] */ BSTR bstrCLSID);

    /* [id] */ HRESULT ( STDMETHODCALLTYPE *UnregisterComponent )(
        IComponentRegistrar * This,
        /* [in] */ BSTR bstrCLSID);

    END_INTERFACE
} IComponentRegistrarVtbl;

interface IComponentRegistrar
{
    CONST_VTBL struct IComponentRegistrarVtbl *lpVtbl;
};

#ifndef COBJMACROS

#define IComponentRegistrar_QueryInterface(This,riid,ppvObject) \
    ((This->lpVtbl->QueryInterface)(This,riid,ppvObject))

#define IComponentRegistrar_AddRef(This) \
    ((This->lpVtbl->AddRef)(This))

#define IComponentRegistrar_Release(This) \
    ((This->lpVtbl->Release)(This))

#define IComponentRegistrar_GetTypeInfoCount(This,pctinfo) \
    ((This->lpVtbl->GetTypeInfoCount)(This,pctinfo))

#define IComponentRegistrar_GetTypeInfo(This,iTInfo,lcid,ppTInfo) \
    ((This->lpVtbl->GetTypeInfo)(This,iTInfo,lcid,ppTInfo))

#define IComponentRegistrar_GetIDsOfNames(This,riid,rgszNames,cNames,lcid,rgDispId) \
    ((This->lpVtbl->GetIDsOfNames)(This,riid,rgszNames,cNames,lcid,rgDispId))

#define IComponentRegistrar_Invoke(This,dispIdMember,riid,lcid,wFlags,pDispParams,pVarResult,pExcepInfo,puArgErr) \
    ((This->lpVtbl->Invoke)(This,dispIdMember,riid,lcid,wFlags,pDispParams,pVarResult,pExcepInfo,puArgErr))

#define IComponentRegistrar_Attach(This,bstrPath) \
    ((This->lpVtbl->Attach)(This,bstrPath))

#define IComponentRegistrar_RegisterAll(This) \
    ((This->lpVtbl->RegisterAll)(This))

#define IComponentRegistrar_UnregisterAll(This) \
    ((This->lpVtbl->UnregisterAll)(This))

#define IComponentRegistrar_GetComponents(This,pbastrCLSIDs,pbstrDescriptions) \
    ((This->lpVtbl->GetComponents)(This,pbastrCLSIDs,pbstrDescriptions))

#define IComponentRegistrar_RegisterComponent(This,bstrCLSID) \
    ((This->lpVtbl->RegisterComponent)(This,bstrCLSID))

#define IComponentRegistrar_UnregisterComponent(This,bstrCLSID) \
    ((This->lpVtbl->UnregisterComponent)(This,bstrCLSID))

#endif /* COBJMACROS */

#endif /* C style interface */

```

```

#endif /* __IComponentRegistrar_INTERFACE_DEFINED__ */

#ifdef __tpcc_com_INTERFACE_DEFINED__
#define __tpcc_com_INTERFACE_DEFINED__

/* interface Itpcc_com */
/* [unique][helpstring][uuid][object] */

EXTERN_C const IID IID_Itpcc_com;

#if defined(__cplusplus) && !defined(CINTERFACE)

MIDL_INTERFACE("5B4FA473-2E68-4D79-A626-F38B30B8196E")
Itpcc_com : public IUnknown
{
public:
    virtual /* [helpstring] */ HRESULT STDMETHODCALLTYPE doStockLevel(
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer) = 0;

    virtual /* [helpstring] */ HRESULT STDMETHODCALLTYPE doNewOrder(
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer) = 0;

    virtual /* [helpstring] */ HRESULT STDMETHODCALLTYPE doPayment(
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer) = 0;

    virtual /* [helpstring] */ HRESULT STDMETHODCALLTYPE doOrderStatus(
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer) = 0;

    virtual /* [helpstring] */ HRESULT STDMETHODCALLTYPE doDBInfo( void) = 0;

    virtual /* [helpstring] */ HRESULT STDMETHODCALLTYPE doSetComplete( void) = 0;

};

#else /* C style interface */

typedef struct Itpcc_comVtbl
{
    BEGIN_INTERFACE

    HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
        Itpcc_com * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */
        __RPC__deref_out void **ppvObject);

    ULONG ( STDMETHODCALLTYPE *AddRef )(
        Itpcc_com * This);

    ULONG ( STDMETHODCALLTYPE *Release )(
        Itpcc_com * This);

    /* [helpstring] */ HRESULT ( STDMETHODCALLTYPE *doStockLevel )(
        Itpcc_com * This,
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer);

    /* [helpstring] */ HRESULT ( STDMETHODCALLTYPE *doNewOrder )(
        Itpcc_com * This,
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer);

    /* [helpstring] */ HRESULT ( STDMETHODCALLTYPE *doPayment )(

```

```

        Itpcc_com * This,
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer);

    /* [helpstring] */ HRESULT ( STDMETHODCALLTYPE *doOrderStatus )(
        Itpcc_com * This,
        /* [in] */ INT *size,
        /* [size_is][size_is][out][in] */ UCHAR **buffer);

    /* [helpstring] */ HRESULT ( STDMETHODCALLTYPE *doDBInfo )(
        Itpcc_com * This);

    /* [helpstring] */ HRESULT ( STDMETHODCALLTYPE *doSetComplete )(
        Itpcc_com * This);

    END_INTERFACE
} Itpcc_comVtbl;

interface Itpcc_com
{
    CONST_VTBL struct Itpcc_comVtbl *lpVtbl;
};

#ifdef COBJMACROS

#define Itpcc_com_QueryInterface(This,riid,ppvObject) \
    ((This->lpVtbl->QueryInterface)(This,riid,ppvObject))

#define Itpcc_com_AddRef(This) \
    ((This->lpVtbl->AddRef)(This))

#define Itpcc_com_Release(This) \
    ((This->lpVtbl->Release)(This))

#define Itpcc_com_doStockLevel(This,size,buffer) \
    ((This->lpVtbl->doStockLevel)(This,size,buffer))

#define Itpcc_com_doNewOrder(This,size,buffer) \
    ((This->lpVtbl->doNewOrder)(This,size,buffer))

#define Itpcc_com_doPayment(This,size,buffer) \
    ((This->lpVtbl->doPayment)(This,size,buffer))

#define Itpcc_com_doOrderStatus(This,size,buffer) \
    ((This->lpVtbl->doOrderStatus)(This,size,buffer))

#define Itpcc_com_doDBInfo(This) \
    ((This->lpVtbl->doDBInfo)(This))

#define Itpcc_com_doSetComplete(This) \
    ((This->lpVtbl->doSetComplete)(This))

#endif /* COBJMACROS */

#ifdef /* C style interface */

#endif

#endif /* __tpcc_com_INTERFACE_DEFINED__ */

/* library tpccComLib */
/* [custom][helpstring][version][uuid] */

```

```

EXTERN_C const IID LIBID_tpccComLib;

EXTERN_C const CLSID CLSID_CompReg;

#ifdef __cplusplus

class DECLSPEC_UUID("90EEDAFF-F8D3-4711-99A9-8AC30FE5DB9")
CompReg;
#endif

EXTERN_C const CLSID CLSID_tpcc_com;

#ifdef __cplusplus

class DECLSPEC_UUID("5F752BF2-F739-43D4-8492-44C19581C0A1")
tpcc_com;
#endif
#endif /* __tpccComLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

unsigned long __RPC_USER BSTR_UserSize( unsigned long *, unsigned long , BSTR
*);
unsigned char * __RPC_USER BSTR_UserMarshal( unsigned long *, unsigned char *, BSTR * );
unsigned char * __RPC_USER BSTR_UserUnmarshal(unsigned long *, unsigned char *, BSTR * );
void __RPC_USER BSTR_UserFree( unsigned long *, BSTR * );

unsigned long __RPC_USER LPSAFEARRAY_UserSize( unsigned long *, unsigned
long , LPSAFEARRAY * );
unsigned char * __RPC_USER LPSAFEARRAY_UserMarshal( unsigned long *, unsigned char *,
LPSAFEARRAY * );
unsigned char * __RPC_USER LPSAFEARRAY_UserUnmarshal(unsigned long *, unsigned char *,
LPSAFEARRAY * );
void __RPC_USER LPSAFEARRAY_UserFree( unsigned long *, LPSAFEARRAY * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

## tpccCom/tpccCom.def

; tpccCom.def : Declares the module parameters.

LIBRARY "tpccCom.DLL"

EXPORTS  
 DllCanUnloadNow PRIVATE  
 DllGetClassObject PRIVATE  
 DllRegisterServer PRIVATE  
 DllUnregisterServer PRIVATE

## tpccCom/tpccCom.vcproj

```

<?xml version="1.0" encoding="Windows-1252"?>
<VisualStudioProject
    ProjectType="Visual C++"
    Version="9.00"
    Name="tpccCom"
    ProjectGUID="{3401AE40-CD75-4B1B-8302-6A30DFD4B3B7}"
    RootNamespace="tpccCom"
    Keyword="AtlProj"
    TargetFrameworkVersion="131072"
    >
    <Platforms>

```

```

    <Platform
      Name="Win32"
    />
  </Platforms>
  <ToolFiles>
  </ToolFiles>
  <Configurations>
    <Configuration
      Name="Debug\Win32"
      OutputDirectory="Debug"
      IntermediateDirectory="Debug"
      ConfigurationType="2"
    />
  InheritedPropertySheets="$(VCInstallDir)\VCProjectDefaults\UpgradeFromVC71.vsprops"
  UseOfATL="1"
  ATLMinimizesCRunTimeLibraryUsage="false"
  CharacterSet="2"
  >
  <Tool
    Name="VCPreBuildEventTool"
  />
  <Tool
    Name="VCCustomBuildTool"
  />
  <Tool
    Name="VCXMLDataGeneratorTool"
  />
  <Tool
    Name="VCWebServiceProxyGeneratorTool"
  />
  <Tool
    Name="VCMIDLTool"
    PreprocessorDefinitions="_DEBUG"
    MkTypLibCompatible="false"
    TargetEnvironment="1"
    GenerateStublessProxies="true"
    TypeLibraryName="$(IntDir)\tpccCom.tlb"
    HeaderFileName="tpccCom.h"
    DLLDataFileName=""
    InterfaceIdentifierFileName="tpccCom_i.c"
    ProxyFileName="tpccCom_p.c"
  />
  <Tool
    Name="VCCLCompilerTool"
    Optimization="0"
    AdditionalIncludeDirectories="&quot;C:\home\tpcc_db2\tpcc-
c.ibm\include&quot;;C:\home\tpcc_db2\tpccDB2glue"
    PreprocessorDefinitions="WIN32;_WINDOWS;_USRDLL;_MERGE_PROXYSTUB;DB2;DEBUG"
    MinimalRebuild="false"
    BasicRuntimeChecks="0"
    RuntimeLibrary="2"
    UsePrecompiledHeader="0"
    WarningLevel="3"
    Detect64BitPortabilityProblems="true"
    DebugInformationFormat="4"
  />
  <Tool
    Name="VCManagedResourceCompilerTool"
  />
  <Tool
    Name="VCResourceCompilerTool"
    PreprocessorDefinitions="_DEBUG"
    Culture="1033"
    AdditionalIncludeDirectories="$(IntDir)"
  />
  <Tool
    Name="VCPreLinkEventTool"
  />
  <Tool
    Name="VCLinkerTool"
    IgnoreImportLibrary="true"
    AdditionalOptions="/NODEFAULTLIB:library"
  />

```

```

    AdditionalDependencies="comsvcs.lib"
    OutputFile="$(OutDir)\tpccCom.dll"
    LinkIncremental="2"
    IgnoreDefaultLibraryNames=""
    ModuleDefinitionFile=".tpccCom.def"
    GenerateDebugInformation="true"
    SubSystem="2"
    OptimizeReferences="2"
    EnableCOMDATFolding="2"
    RandomizedBaseAddress="1"
    DataExecutionPrevention="0"
    ImportLibrary="$(OutDir)\tpccCom.lib"
    TargetMachine="1"
  />
  <Tool
    Name="VCALinkTool"
  />
  <Tool
    Name="VCManifestTool"
  />
  <Tool
    Name="VCXDCMakeTool"
  />
  <Tool
    Name="VCBscMakeTool"
  />
  <Tool
    Name="VCFxCopTool"
  />
  <Tool
    Name="VCAppVerifierTool"
  />
  <Tool
    Name="VCPostBuildEventTool"
    Description="Performing registration"
    CommandLine="regsvr32 /s /c &quot;$(TargetPath)&quot;"
  />
  </Configuration>
  <Configuration
    Name="Release\Win32"
    OutputDirectory="Release"
    IntermediateDirectory="Release"
    ConfigurationType="2"
  />
  InheritedPropertySheets="$(VCInstallDir)\VCProjectDefaults\UpgradeFromVC71.vsprops"
  UseOfMFC="1"
  UseOfATL="1"
  ATLMinimizesCRunTimeLibraryUsage="false"
  CharacterSet="2"
  >
  <Tool
    Name="VCPreBuildEventTool"
  />
  <Tool
    Name="VCCustomBuildTool"
  />
  <Tool
    Name="VCXMLDataGeneratorTool"
  />
  <Tool
    Name="VCWebServiceProxyGeneratorTool"
  />
  <Tool
    Name="VCMIDLTool"
    PreprocessorDefinitions="NDEBUG"
    MkTypLibCompatible="false"
    TargetEnvironment="1"
    GenerateStublessProxies="true"
    TypeLibraryName="$(IntDir)\tpccCom.tlb"
    HeaderFileName="tpccCom.h"
    DLLDataFileName=""
    InterfaceIdentifierFileName="tpccCom_i.c"
    ProxyFileName="tpccCom_p.c"
  />

```

```

  />
  <Tool
    Name="VCCLCompilerTool"
    Optimization="2"
    InlineFunctionExpansion="0"
    EnableIntrinsicFunctions="false"
    FavorSizeOrSpeed="1"
    OmitFramePointers="true"
    EnableFiberSafeOptimizations="false"
    AdditionalIncludeDirectories=".tpc-c.ibm\include;.tpccDB2glue"
  />
  PreprocessorDefinitions="WIN32;_WINDOWS;NDEBUG;_USRDLL;_MERGE_PROXYSTUB;DB2;
RT_DEBUG"
  RuntimeLibrary="0"
  UsePrecompiledHeader="0"
  WarningLevel="3"
  Detect64BitPortabilityProblems="true"
  DebugInformationFormat="3"
  />
  <Tool
    Name="VCManagedResourceCompilerTool"
  />
  <Tool
    Name="VCResourceCompilerTool"
    PreprocessorDefinitions="NDEBUG"
    Culture="1033"
    AdditionalIncludeDirectories="$(IntDir)"
  />
  <Tool
    Name="VCPreLinkEventTool"
  />
  <Tool
    Name="VCLinkerTool"
    IgnoreImportLibrary="true"
    AdditionalOptions="/NODEFAULTLIB:library"
    AdditionalDependencies="comsvcs.lib"
    OutputFile="$(OutDir)\tpccCom.dll"
    LinkIncremental="1"
    ModuleDefinitionFile=".tpccCom.def"
    GenerateDebugInformation="true"
    SubSystem="2"
    OptimizeReferences="2"
    EnableCOMDATFolding="2"
    RandomizedBaseAddress="1"
    DataExecutionPrevention="0"
    ImportLibrary="$(OutDir)\tpccCom.lib"
    TargetMachine="1"
  />
  <Tool
    Name="VCALinkTool"
  />
  <Tool
    Name="VCManifestTool"
    EmbedManifest="true"
  />
  <Tool
    Name="VCXDCMakeTool"
  />
  <Tool
    Name="VCBscMakeTool"
  />
  <Tool
    Name="VCFxCopTool"
  />
  <Tool
    Name="VCAppVerifierTool"
  />
  <Tool
    Name="VCPostBuildEventTool"
    Description="Performing registration"
    CommandLine="regsvr32 /s /c &quot;$(TargetPath)&quot;"
  />
  </Configuration>

```

```

</Configurations>
<References>
</References>
<Files>
  <Filter
    Name="Source Files"
    Filter="*.cpp;*.c;*.cxx;*.def;*.odj;*.idl;*.h;*.hpp;*.bat;*.asm;*.asmx"
    UniqueIdentifier="{4FC737F1-C7A5-4376-A066-2A32D752A2FF}"
  >
    <File
      RelativePath=".lcompreg.cpp"
    >
  </File>
  <File
    RelativePath=".dllldatax.c"
  >
    <FileConfiguration
      Name="Debug|Win32"
    >
      <Tool
        Name="VCCLCompilerTool"
        UsePrecompiledHeader="0"
      >
    </Tool>
  </FileConfiguration>
  <FileConfiguration
    Name="Release|Win32"
  >
    <Tool
      Name="VCCLCompilerTool"
      UsePrecompiledHeader="0"
    >
  </Tool>
  </FileConfiguration>
  </File>
  <File
    RelativePath=".lstdafx.cpp"
  >
    <FileConfiguration
      Name="Debug|Win32"
    >
      <Tool
        Name="VCCLCompilerTool"
        UsePrecompiledHeader="1"
      >
    </Tool>
  </FileConfiguration>
  <FileConfiguration
    Name="Release|Win32"
  >
    <Tool
      Name="VCCLCompilerTool"
      UsePrecompiledHeader="1"
    >
  </Tool>
  </FileConfiguration>
  </File>
  <File
    RelativePath="..tpccsapitime.cpp"
  >
  </File>
  <File
    RelativePath=".tpcc_com.cpp"
  >
  </File>
  <File
    RelativePath=".tpccCom.cpp"
  >
  </File>
  <File
    RelativePath=".tpccCom.def"
  >
  </File>
  <File
    RelativePath=".tpccCom.idl"
  >
  </File>
  </Filter>

```

```

</Filter>
<Filter
  Name="Header Files"
  Filter="*.h;*.hpp;*.hxx;*.hm;*.inl;*.inc;*.xsd"
  UniqueIdentifier="{933995380-89BD-4b04-88EB-625FBE52EBFB}"
>
  <File
    RelativePath=".lcompreg.h"
  >
  </File>
  <File
    RelativePath=".dllldatax.h"
  >
  </File>
  <File
    RelativePath=".lResource.h"
  >
  </File>
  <File
    RelativePath=".lstdafx.h"
  >
  </File>
  <File
    RelativePath=".tpcc_com.h"
  >
  </File>
</Filter>
<Filter
  Name="Resource Files"
  Filter="*.rc;*.ico;*.cur;*.bmp;*.dlg;*.rc2;*.rct;*.bin;*.rgs;*.gif;*.jpg;*.jpeg;*.jpe;*.resx"
  UniqueIdentifier="{67DA6AB6-F800-4c08-8B7A-83BB121AAD01}"
>
  <File
    RelativePath=".tpcc_com.rgs"
  >
  </File>
  <File
    RelativePath=".tpccCom.rc"
  >
  </File>
  <File
    RelativePath=".tpccCom.rgs"
  >
  </File>
</Filter>
<Filter
  Name="Generated Files"
  SourceControlFiles="false"
>
  <File
    RelativePath=".tpccCom_i.c"
  >
    <FileConfiguration
      Name="Debug|Win32"
    >
      <Tool
        Name="VCCLCompilerTool"
        UsePrecompiledHeader="0"
      >
    </Tool>
  </FileConfiguration>
  <FileConfiguration
    Name="Release|Win32"
  >
    <Tool
      Name="VCCLCompilerTool"
      UsePrecompiledHeader="0"
    >
  </Tool>
  </FileConfiguration>
  </File>
</Filter>
<File
  RelativePath=".lReadMe.txt"
>

```

```

  </File>
  <Files>
  <Globals>
  </Globals>
</VisualStudioProject>

tpccCom/tpccCom.rc

// Microsoft Visual C++ generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "winres.h"

//
// English (U.S.) resources
//

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#include _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

#ifdef APSTUDIO_INVOKED
//
// TEXTINCLUDE
//
1 TEXTINCLUDE
BEGIN
  "resource.h\0"
END

2 TEXTINCLUDE
BEGIN
  "#include ""winres.h""\r\n"
  "\0"
END

3 TEXTINCLUDE
BEGIN
  "1 TYPELIB ""tpccCom.tlb""\r\n"
  "\0"
END

#endif // APSTUDIO_INVOKED

//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x01L
#else
FILEFLAGS 0x00L
#endif

```



```

FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
  BLOCK "StringFileInfo"
  BEGIN
    BLOCK "040904e4"
    BEGIN
      VALUE "CompanyName", "TODO: <Company name>"
      VALUE "FileDescription", "TODO: <File description>"
      VALUE "FileVersion", "1.0.0.1"
      VALUE "LegalCopyright", "TODO: (c) <Company name>. All rights reserved."
      VALUE "InternalName", "tpccCom.dll"
      VALUE "OriginalFilename", "tpccCom.dll"
      VALUE "ProductName", "TODO: <Product name>"
      VALUE "ProductVersion", "1.0.0.1"
    END
  END
  BLOCK "VarFileInfo"
  BEGIN
    VALUE "Translation", 0x409, 1252
  END
END

```

```

////////////////////////////////////
//
// REGISTRY
//
IDR_TPCCCOM     REGISTRY     "tpccCom.rgs"
IDR_TPCC_COM    REGISTRY     "tpcc_com.rgs"

```

```

////////////////////////////////////
//
// String Table
//

```

```

STRINGTABLE
BEGIN
  IDS_PROJNAME     "tpccCom"
END

```

```

#endif // English (U.S.) resources
////////////////////////////////////

```

```

#ifndef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpccCom.tlb"

```

```

////////////////////////////////////
#endif // not APSTUDIO_INVOKED

```

### tpccCom/tpcc\_com.rgs

```

HKCR
{
  tpccCom.tpcc_com.1 = s 'tpcc_com Class'
  {
    CLSID = s '{5F752BF2-F739-43D4-8492-44C19581C0A1}'
  }
  tpccCom.tpcc_com = s 'tpcc_com Class'
  {
    CLSID = s '{5F752BF2-F739-43D4-8492-44C19581C0A1}'
    CurVer = s 'tpccCom.tpcc_com.1'
  }
}

```

```

NoRemove CLSID
{
  ForceRemove (5F752BF2-F739-43D4-8492-44C19581C0A1) = s 'tpcc_com Class'
  {
    ProgID = s 'tpccCom.tpcc_com.1'
    VersionIndependentProgID = s 'tpccCom.tpcc_com'
    InprocServer32 = s '%MODULE%'
    {
      val ThreadingModel = s 'Both'
    }
    val AppID = s '%APPID%'
    "TypeLib" = s '{91F1B8B0-89E9-457B-A228-3E2D6CE3E752}'
  }
}

```

### tpccCom/tpccCom.rgs

```

HKCR
{
  NoRemove AppID
  {
    "%APPID%" = s 'tpccCom'
    'tpccCom.DLL'
    {
      val AppID = s '%APPID%'
    }
  }
  NoRemove CLSID
  {
    ForceRemove (90EEDAFF-F8D3-4711-99A9-8AC3C0FE5DB9) = s 'CompReg Class'
    {
      InprocServer32 = s '%MODULE%'
      {
        val ThreadingModel = s 'Apartment'
      }
      "TypeLib" = s '{91F1B8B0-89E9-457B-A228-3E2D6CE3E752}'
    }
  }
}

```

### tpccDB2glue/stdafx.h

```

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

```

```
#pragma once
```

```

#define WIN32_LEAN_AND_MEAN // Exclude rarely-used stuff from Windows headers
// Windows Header Files:
#include <windows.h>

```

```
// TODO: reference additional headers your program requires here
```

### tpccDB2glue/stdafx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// tpccDB2glue.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

```

```
#include "stdafx.h"
```

```

// TODO: reference any additional headers you need in STDAFX.H
// and not in this file

```

### tpccDB2glue/time.cpp

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/timeb.h>
#include <time.h>

char *get_time_prefix(char *buffer)
{
  time_t cur_time;
  char time_str[30];
  int len;

  cur_time = time(&cur_time);
  strftime(time_str, 29, "%X", localtime(&cur_time));

  len = sprintf(buffer, "%s - ",
               time_str);
  if (len >= 30) {
    sprintf(buffer, "too small: %d\n",
            30, len);
  }
  return(buffer);
}

```

### tpccDB2glue/tpccDB2glue.h

```

// The following ifdef block is the standard way of creating macros which make exporting
// from a DLL simpler. All files within this DLL are compiled with the TPCCDB2GLUE_EXPORTS
// symbol defined on the command line. This symbol should not be defined on any project
// that uses this DLL. This way any other project whose source files include this file see
// TPCCDB2GLUE_API functions as being imported from a DLL, whereas this DLL sees symbols
// defined with this macro as being exported.

```

```

#ifdef TPCCDB2GLUE_EXPORTS
#define TPCCDB2GLUE_API __declspec(dllexport)
#else
#define TPCCDB2GLUE_API __declspec(dllimport)
#endif

```

```

#ifndef SPGENERAL
#define SPGENERAL
#endif

```

```

#include <db2tpcc.h>
#include <tpcc.h>

```

```

////////////////////////////////////
// Error/Debug log file defines
////////////////////////////////////
ofstream debugStream;
ofstream errorStream;
ofstream rtStream;

```

```

CRITICAL_SECTION debugMutex;
CRITICAL_SECTION errorMutex;
CRITICAL_SECTION comMutex;

```

```

#ifdef RT_DEBUG
int count;
#endif

```

```

#ifdef TIMING
FILE *respTimes;
struct txn
{
  short txnType;
  struct_timeb startime;
  struct_timeb endime;
  short padding;
};

```

```

////////////////////////////////////
// Registry Values
////////////////////////////////////
#define DB_USER_NAME          "dbUserName"
#define DB_USER_PASSWORD     "dbPassword"
#define DB_NAME               "dbName"

```

```

char userName[16] = {NULL};
char userPassword[16] = {NULL};

HKEY registryKey;
DWORD regType;
char value[MAX_STRING_LEN];
DWORD regValueSize = MAX_STRING_LEN;

```

```

////////////////////////////////////
// DB2 Glue Function Prototypes
////////////////////////////////////
extern "C" TPCCDB2GLUE_API int connect_db(char *dbName, void **ctx);
extern "C" TPCCDB2GLUE_API int getContext(void **ctx);
extern "C" TPCCDB2GLUE_API int detachContext(void *ctx);
extern "C" TPCCDB2GLUE_API int attachContext(void *ctx);
extern "C" TPCCDB2GLUE_API int disconnect_db(void *ctx);

```

```

extern "C" TPCCDB2GLUE_API int do_nord(nord_wrapper *nord, void *ctx);
extern "C" TPCCDB2GLUE_API int do_pymt(paym_wrapper *pymt, void *ctx);
extern "C" TPCCDB2GLUE_API int do_ords(ords_wrapper *ords, void *ctx);
extern "C" TPCCDB2GLUE_API int do_dvly(dvly_wrapper *dvly, void *ctx);
extern "C" TPCCDB2GLUE_API int do_stok(stok_wrapper *stok, void *ctx);

```

### tpccDB2glue/tpccDB2glue.cpp

```

// tpccDB2glue.cpp : Defines the entry point for the DLL application.

```

```

#include "stdafx.h"
#include "tpccDB2glue.h"

```

```

int debugFlag;

#ifdef RT_DEBUG
#ifdef WIN32
    struct _timeb start_time; // time that the shared memory was initialized
#else
    struct timeb start_time; // time that the shared memory was initialized
#endif
#endif

```

```

#ifdef RT_DEBUG
//
=====
UINT timestamp()
{
#ifdef WIN32
    struct _timeb tb;
    _ftime( &tb );
#else
    struct timeb tb;
    ftime( &tb );
#endif
    // returns time since the SharedMem class was constructed
    return ((tb.time - start_time.time) * 1000) + tb.millitm - start_time.millitm;
}
#endif

```

```

BOOL WINAPI DllMain( HANDLE hModule,
                    DWORD ul_reason_for_call,
                    LPVOID lpReserved

```

```

{

```

```

    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        InitializeCriticalSection(&comMutex);
        if(debugFlag)
        {
            InitializeCriticalSection(&debugMutex);

            debugStream.rdbuf( )
>open("C:\\inetpub\\wwwroot\\tpcc\\debug_gluecode.txt", ios_base::in | ios_base::out | ios_base::app);
            if(!debugStream.rdbuf( )->is_open())
                return FALSE;
        }

        DEBUGMSG("Entered dllMain of tpccDB2glue.dll" << endl);
        InitializeCriticalSection(&errorMutex);
        errorStream.rdbuf( )->open("C:\\inetpub\\wwwroot\\tpcc\\error_gluecode.txt", ios_base::in |
ios_base::out);
        if(!errorStream.rdbuf( )->is_open())
            return FALSE;

        DEBUGMSG("Opening registry sub key "<< REGISTRY_SUB_KEY << endl);
        //open up registry key
        if(RegOpenKeyEx(HKEY_LOCAL_MACHINE, REGISTRY_SUB_KEY, 0, KEY_READ, &registryKey)
== ERROR_SUCCESS)
        {
            DEBUGMSG("Registry key open" << endl);
            //get the null db user name
            regValueSize = sizeof(value);
            if (RegQueryValueEx(registryKey, DB_USER_NAME, 0, &regType, (BYTE *)
&value, &regValueSize) == ERROR_SUCCESS )
                strcpy(userName, value);
            else
                return ERR_INVALID_USERNAME;
            DEBUGMSG("DB user name:" << userName << endl);

            regValueSize = sizeof(value);
            if (RegQueryValueEx(registryKey, DB_USER_PASSWORD, 0, &regType, (BYTE
*) &value, &regValueSize) == ERROR_SUCCESS )
                strcpy(userPassword, value);
            else
                return ERR_INVALID_PASSWORD;
            DEBUGMSG("DB user password:" << userPassword << endl);
        }
        else
        {
            return ERR_INVALID_REGISTRY_KEY;
            DEBUGMSG("Unable to open registry key" << REGISTRY_SUB_KEY << endl);
        }
        break;
    case DLL_THREAD_ATTACH:
        break;
    case DLL_THREAD_DETACH:
        break;
    case DLL_PROCESS_DETACH:
        #ifdef TIMING
            ERRORMSG("dll_process_detach called, closing timing file" << endl);
            fclose(respTimes);
        #endif
        break;
    }
    return TRUE;
}

/*
*****
** Name : attachContext
** Description :

```

```

** Function calls db2 api to attach thread to
** a specific context per thread basis.
** Parameters : void* stored context
** Returns : int - return code
** Comments :
**
*****
*/
extern "C" int attachContext(void *ctx)
{
    int rc;
    if ( (rc = attach_context(ctx)) != OK)
        return ERR_ATTACHING_CONTEXT;

    return OK;
}

/*
*****
** Name : detachContext
** Description : Function calls db2 api to detach thread from context
** Parameters : void* stored context
** Returns : int - return code
** Comments :
**
*****
*/
extern "C" int detachContext(void *ctx)
{
    int rc;
    if ( (rc = detach_context(ctx)) != OK)
    {
        ERRORMSG("error detaching context from db, rc:" << rc << endl);
        return ERR_DETACHING_CONTEXT;
    }
    return OK;
}

/*
*****
** Name : connect_db
** Description : Function calls db2 api to connect to db
** Parameters : char* dbName
                void** uninitialized context
** Returns : int - return code
** Comments : To connect to db, first connection must be
** established. Next, context for that connect
** be saved off. Finally, detach from the
** context just created.
**
*****
*/
extern "C" TPCCDB2GLUE_API int connect_db(char *dbName, void **ctx)
{
    DEBUGMSG("Entered db2glue do_connect using dbName:" << dbName << endl << "Calling
connect_to_TM_auth() with username:" << userName << " password:" << userPassword << endl);
    char buf[50];
#ifdef RT_DEBUG
        EnterCriticalSection(&comMutex);
        count++;
        sprintf(buf, "C:\\inetpub\\wwwroot\\tpcc\\db2_rt_%d", count);
        LeaveCriticalSection(&comMutex);
        rtStream.rdbuf( )->open(buf, ios_base::in | ios_base::out);
        if(!respTimes)

```

```

        {
            ERRORMSG("Unable to open response time file
c:\inetpub\wwwroot\tpcc\respTimes"<<endl);
            return FALSE;
        }
        ERRORMSG("Response time file created:"<<endl);
#ifdef WIN32
        _ftime(&start_time); // get the time when shared memory is created
#else
        ftime(&start_time); // get the time when shared memory is created
#endif
        #endif
        int rc = connect_to_TM_auth(dbName,userName,userPassword);
        if(rc != OK)
        {
            DEBUGMSG("Object do_connect failed, rc:"<<rc<<endl);
            ERRORMSG("Object do_connect failed, rc:"<<rc<<endl);
        }
        return rc;
    }
    DEBUGMSG("calling get_context"<<endl);
    if ( (rc = get_context(ctx)) != OK)
    {
        DEBUGMSG("Object get_context() failed, rc:"<<rc <<endl);
        ERRORMSG("Object get_context() failed, rc:"<<rc <<endl);
    }
    return ERR_SAVING_CONTEXT;
}
DEBUGMSG("Object get_context successful, context:"<< DEBUGADDRESS("ctx)<<
saved"<<endl);
DEBUGMSG("Object calling detach_context() w/ ctx:"<<DEBUGADDRESS("ctx)<<endl);
if ( (rc = detach_context("ctx)) != OK)
{
    DEBUGMSG("Object failed detach_context w/ ctx:"<<DEBUGADDRESS("ctx)<<" rc:" <<
rc <<endl);
    ERRORMSG("Object failed detach_context w/ ctx:"<<DEBUGADDRESS("ctx)<<" rc:" <<
rc <<endl);
    return ERR_DETACHING_CONTEXT;
}
DEBUGMSG("Object detach_context successful, context:"<<DEBUGADDRESS("ctx)<< ",
connection complete"<<endl);
return OK;
}
/*
*****
** Name : disconnect_db
** Description : Function calls db2 api to disconnect from db
** Parameters : void* stored context
** Returns : int - return code
** Comments : To disconnect from db, first must attach to
thread's context. Next, disconnect from db
*****
*/
extern "C" TPCCDB2GLUE_API int disconnect_db(void *ctx)
{
    DEBUGMSG("Entered do_disconnect, attaching to context:" << DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc <<
endl);
        DEBUGMSG("failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc <<
endl);
    }
}

```

```

        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("context established. preparing to call db2" << endl);
    rc = disconnect_from_TM();
    if(rc != OK)
    {
        DEBUGMSG("disconnect failed, rc:"<<rc<<endl);
        ERRORMSG("disconnect failed, rc:"<<rc<<endl);
        return rc;
    }
    return OK;
}
/*
*****
** Name : do_nord
** Description : Function calls db2 api to execute nord txn
** Parameters : nord_wrapper* new order txn structs wrapper
void* stored context
** Returns : int - return code
** Comments : Attach to thread's context, call nord sql function
then detach from context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_nord(nord_wrapper *nord,void *ctx)
{
#ifdef RT_DEBUG
    struct _ftime times[4];
    _ftime(&times[0]);
#endif
    DEBUGMSG("Entered do_nord, attaching to context:" << DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("nord failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
        DEBUGMSG("nord failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attached to context:" << DEBUGADDRESS(ctx)<< ", preparing to call db2" << endl);
#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif
    //call new order txn
#ifdef RT_DEBUG
    _ftime(&times[1]);
#endif
    nord->out_nord.s_transtatus = neword_sql(&nord->in_nord,&nord->out_nord);
#ifdef RT_DEBUG
    _ftime(&times[2]);
#endif
#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=1;
    EnterCriticalSection(&errorMutex);
    rc = fwrite(&timeSample,sizeof(struct txn),1,respTimes);
    LeaveCriticalSection(&errorMutex);
#endif
    DEBUGMSG("return from neword_sql(), s_transtatus:" << nord->out_nord.s_transtatus << endl);
}

```

```

        rc = detachContext(ctx);
        if(rc != OK)
        {
            ERRORMSG("nord failed detach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
            DEBUGMSG("nord failed detach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
        }
        return ERR_DETACHING_CONTEXT;
    }
#ifdef RT_DEBUG
    _ftime(&times[3]);
    RTMSG("0 - " << nord->in_nord.s_W_ID << " " << nord->in_nord.s_D_ID << " " << nord-
>out_nord.s_O_ID << " " <<times[0].time << " " << times[0].millitm << " " << times[1].time << " " <<
times[1].millitm << " " << times[2].time << " " << times[2].millitm << " " << times[3].time << " " <<
times[3].millitm << " " << endl);
#endif
    return OK;
}
/*
*****
** Name : do_pymt
** Description : Function calls db2 api to execute pymt txn
** Parameters : paym_wrapper* payment txn structs wrapper
void* stored context
** Returns : int - return code
** Comments : Attach to thread's context, call nord sql function
then detach from context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_pymt(paym_wrapper *pymt,void *ctx)
{
    DEBUGMSG("Entered do_pymt, attaching to context:" << DEBUGADDRESS(ctx) << endl);
#ifdef RT_DEBUG
    struct _ftime times[4];
    _ftime(&times[0]);
#endif
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("pymt failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
        DEBUGMSG("pymt failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
        return ERR_ATTACHING_CONTEXT;
    }
    DEBUGMSG("attached to context:"<< DEBUGADDRESS(ctx) <<" preparing to call db2" << endl);
#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif
    //call pymt txn
#ifdef RT_DEBUG
    _ftime(&times[1]);
#endif
    pymt->out_paym.s_transtatus = payment_sql(&pymt->in_paym,&pymt->out_paym);
#ifdef RT_DEBUG
    _ftime(&times[2]);
#endif
#ifdef TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=2;
    EnterCriticalSection(&errorMutex);
}

```

```

    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1)
    {
        ERRORMSG("Unable to write to binary file, pymt"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif
DEBUGMSG("return from payment_sql(), s_transtatus:" << pymt->out_paym.s_transtatus <<
endl);
rc = detachContext(ctx);
if(rc != OK)
{
    ERRORMSG("pymt failed detach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<endl);
    DEBUGMSG("pymt failed detach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
}
return ERR_DETACHING_CONTEXT;
}
DEBUGMSG("pymt detach_context successful. pymt txn complete."<<endl);

#endif RT_DEBUG
    _ftime(&times[3]);
    RTMSG("1 - " << pymt->in_paym.s_W_ID << " " << pymt->in_paym.s_D_ID << " " << pymt-
>out_paym.s_C_BALANCE << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " " <<
times[1].millitm << " " << times[2].time << " " << times[2].millitm << " " << times[3].time << " " <<
times[3].millitm << " " << endl);
#endif
return OK;
}

/*
*****
** Name          :      do_ords
** Description   :      Function calls db2 api to execute ords txn
** Parameters    :      ords_wrapper*  order status txn structs wrapper
                        void*          stored context
** Returns      :      int - return code
** Comments     :      Attach to thread's context, call nord sql function
                        then detach from context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_ords(ords_wrapper *ords,void *ctx)
{
#endif RT_DEBUG
    struct _timeb times[4];
    _ftime(&times[0]);
#endif
DEBUGMSG("Entered do_ords, attaching to context:" << DEBUGADDRESS(ctx) << endl);
int rc = attachContext(ctx);
if(rc != OK)
{
    ERRORMSG("ords failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
    DEBUGMSG("ords failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
}
return ERR_ATTACHING_CONTEXT;
}
DEBUGMSG("attached to context:"<<DEBUGADDRESS(ctx)<<" , preparing to call db2" << endl);
DEBUGMSG("calling ordstat_sql()" << endl);

#endif TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif
#endif RT_DEBUG
    _ftime(&times[1]);

```

```

#endif
ords->out_ords.s_transtatus = ordstat_sql(&ords->in_ords,&ords->out_ords);
#endif RT_DEBUG
    _ftime(&times[2]);
#endif
#endif TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=3;
    EnterCriticalSection(&errorMutex);
    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1)
    {
        ERRORMSG("Unable to write to binary file, ords"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif
DEBUGMSG("return from ordstat_sql(), s_transtatus:" << ords->out_ords.s_transtatus << endl);
rc = detachContext(ctx);
if(rc != OK)
{
    ERRORMSG("ords failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
    DEBUGMSG("ords failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
}
return ERR_DETACHING_CONTEXT;
}
DEBUGMSG("ords detach_context successful. pymt txn complete."<<endl);

#endif RT_DEBUG
    _ftime(&times[3]);
    RTMSG("2 - " << ords->in_ords.s_W_ID << " " << ords->in_ords.s_D_ID << ords-
>out_ords.s_C_BALANCE << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " " <<
times[1].millitm << " " << times[2].time << " " << times[2].millitm << " " << times[3].time << " " <<
times[3].millitm << " " << endl);
#endif
return OK;
}

/*
*****
** Name          :      do_dlvly
** Description   :      Function calls db2 api to execute ords txn
** Parameters    :      dlvly_wrapper*  dlvly txn structs wrapper
                        void*          stored context
** Returns      :      int - return code
** Comments     :      Attach to thread's context, call nord sql function
                        then detach from context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_dlvly(dlvly_wrapper *dlvly,void *ctx)
{
#endif RT_DEBUG
    struct _timeb times[4];
    _ftime(&times[0]);
#endif
DEBUGMSG("Entered do_dlvly, attaching to context:" << DEBUGADDRESS(ctx) << endl);
int rc = attachContext(ctx);
if(rc != OK)
{
    ERRORMSG("dlvly failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
    DEBUGMSG("dlvly failed attach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc <<
endl);
}
return ERR_ATTACHING_CONTEXT;
}
DEBUGMSG("attached to context:"<<DEBUGADDRESS(ctx)<<" , preparing to call db2" << endl);

```

```

    DEBUGMSG("calling delivery_sql" << endl);
#endif TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif
//call dlvly txn
#endif RT_DEBUG
    _ftime(&times[1]);
#endif
dlvly->out_dlvly.s_transtatus = delivery_sql(&dlvly->in_dlvly,&dlvly->out_dlvly);
#endif RT_DEBUG
    _ftime(&times[2]);
#endif
#endif TIMING
    _ftime(&timeSample.endTime);
    timeSample.txnType=3;
    EnterCriticalSection(&errorMutex);
    if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) != 1)
    {
        ERRORMSG("Unable to write to binary file, dlvly"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif
DEBUGMSG("return from delivery_sql(), s_transtatus:" << dlvly->out_dlvly.s_transtatus << endl);
rc = detachContext(ctx);
if(rc != OK)
{
    ERRORMSG("dlvly failed detach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
    DEBUGMSG("dlvly failed detach_context w/ ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc
<< endl);
}
return ERR_DETACHING_CONTEXT;
}
DEBUGMSG("dlvly detach_context successful. dlvly txn complete."<<endl);
#endif RT_DEBUG
    _ftime(&times[3]);
    RTMSG("4 - " << dlvly->in_dlvly.s_W_ID << " " << dlvly->in_dlvly.s_O_CARRIER_ID << " " << dlvly-
>out_dlvly.s_O_ID << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " " <<
times[1].millitm << " " << times[2].time << " " << times[2].millitm << " " << times[3].time << " " <<
times[3].millitm << " " << endl);
#endif
return OK;
}

/*
*****
** Name          :      do_stok
** Description   :      Function calls db2 api to execute stok txn
** Parameters    :      stok_wrapper*  stok txn structs wrapper
                        void*          stored context
** Returns      :      int - return code
** Comments     :      Attach to thread's context, call nord sql function
                        then detach from context.
*****
*/
extern "C" TPCCDB2GLUE_API int do_stok(stok_wrapper *stok,void *ctx)
{
#endif RT_DEBUG
    struct _timeb times[4];
    _ftime(&times[0]);
#endif

```

```

DEBUGMSG("Entered do_stok, attaching to context:" << DEBUGADDRESS(cb) << endl);
int rc = attachContext(cb);
if(rc != OK)
{
    ERRORMSG("stok failed attach_context w/ cb:" << DEBUGADDRESS(cb) << " rc:" << rc << endl);
    DEBUGMSG("stok failed attach_context w/ cb:" << DEBUGADDRESS(cb) << " rc:" << rc << endl);
}
return ERR_ATTACHING_CONTEXT;
}
DEBUGMSG("attaching to context:" << DEBUGADDRESS(cb) << " preparing to call db2" << endl);
DEBUGMSG("calling stocklev_sq()" << endl);
#endif TIMING
struct txn timeSample;
_time(&timeSample.startTime);
#endif
//call stock level txn
#endif RT_DEBUG
_time(&times[1]);
#endif
stok->out_stok.s_transtatus = stocklev_sq(&stok->in_stok, &stok->out_stok);
#endif RT_DEBUG
_time(&times[2]);
#endif
#endif TIMING
_time(&timeSample.endTime);
timeSample.txnType=5;
EnterCriticalSection(&errorMutex);
if( (fwrite(&timeSample,sizeof(struct txn),1,resTimes)) != 1 )
{
    ERRORMSG("Unable to write to binary file, stok" << endl);
}
LeaveCriticalSection(&errorMutex);
#endif
DEBUGMSG("return from stocklev_sq(), s_transtatus:" << stok->out_stok.s_transtatus << endl);
DEBUGMSG("calling detach_context" << endl);
rc = detachContext(cb);
if(rc != OK)
{
    ERRORMSG("stok failed attach_context w/ cb:" << DEBUGADDRESS(cb) << " rc:" << rc << endl);
    DEBUGMSG("stok failed attach_context w/ cb:" << DEBUGADDRESS(cb) << " rc:" << rc << endl);
}
return ERR_DETACHING_CONTEXT;
}
DEBUGMSG("detach_context successful. stok txn complete." << endl);
#endif RT_DEBUG
_time(&times[3]);
RTMSG("3 - " << stok->in_stok.s_W_ID << " " << stok->in_stok.s_D_ID << " " << stok->out_stok.s_low_stock << " " << times[0].time << " " << times[0].millitm << " " << times[1].time << " " << times[1].millitm << " " << times[2].time << " " << times[2].millitm << " " << times[3].time << " " << times[3].millitm << " " << endl);
#endif
return OK;
}

```

### tpccDB2glue/tpccDB2glue.vcproj

```

<?xml version="1.0" encoding="Windows-1252"?>
<VisualStudioProject
ProjectType="Visual C++"
Version="9.00"

```

```

Name="tpccDB2glue"
ProjectGUID="{00B934F6-BE49-4022-AF4A-4A785DF42AE}"
RootNamespace="tpccDB2glue"
Keyword="Win32Proj"
TargetFrameworkVersion="131072"
>
<Platforms>
  <Platform
    Name="Win32"
  />
</Platforms>
<ToolFiles>
</ToolFiles>
<Configurations>
  <Configuration
    Name="Debug|Win32"
    OutputDirectory="Debug"
    IntermediateDirectory="Debug"
    ConfigurationType="2"
  />
  InheritedPropertySheets="$ (VCInstallDir)VCProjectDefaults\UpgradeFromVC71.vsprops"
  UseOfMFC="1"
  CharSet="2"
  <Tool
    Name="VCPreBuildEventTool"
  />
  <Tool
    Name="VCCustomBuildTool"
  />
  <Tool
    Name="VCXMLDataGeneratorTool"
  />
  <Tool
    Name="VCWebServiceProxyGeneratorTool"
  />
  <Tool
    Name="VCMDLTool"
  />
  <Tool
    Name="VCCLCompilerTool"
    Optimization="0"
    AdditionalIncludeDirectories="..\tpccsapi\&quot;C:\home\tpcc_db2\tpcc-
c.ibm\include&quot;;"
  />
  PreprocessorDefinitions="WIN32;_DEBUG;_WINDOWS;_USRDLL;TPCCDB2GLUE_EXPORTS"
  MinimalRebuild="true"
  BasicRuntimeChecks="3"
  RuntimeLibrary="1"
  StructMemberAlignment="0"
  UsePrecompiledHeader="0"
  WarningLevel="3"
  Detect64BitPortabilityProblems="true"
  DebugInformationFormat="4"
  />
  <Tool
    Name="VCManagedResourceCompilerTool"
  />
  <Tool
    Name="VCResourceCompilerTool"
  />
  <Tool
    Name="VCPreLinkEventTool"
  />
  <Tool
    Name="VCLinkerTool"
    AdditionalDependencies="C:\home\tpcc_db2\tpcc-c.ibm\Src.Cil\tpcccli.lib
&quot;c:\IBM\SQLLIB\lib\db2api.lib&quot;; Advapi32.lib"
    OutputFile="$ (OutDir)\tpccDB2glue.dll"
    LinkIncremental="2"
    AdditionalLibraryDirectories="c:\windows\system32"
    IgnoreAllDefaultLibraries="false"
    GenerateDebugInformation="true"
  />

```

```

ProgramDatabaseFile="$ (OutDir)\tpccDB2glue.pdb"
SubSystem="2"
RandomizedBaseAddress="1"
DataExecutionPrevention="0"
ImportLibrary="$ (OutDir)\tpccDB2glue.lib"
TargetMachine="1"
/>
<Tool
  Name="VCALinkTool"
/>
<Tool
  Name="VCManifestTool"
/>
<Tool
  Name="VCXDCMakeTool"
/>
<Tool
  Name="VCBscMakeTool"
/>
<Tool
  Name="VCFxCopTool"
/>
<Tool
  Name="VCAppVerifierTool"
/>
<Tool
  Name="VCPostBuildEventTool"
/>
</Configuration>
<Configuration
  Name="Release|Win32"
  OutputDirectory="Release"
  IntermediateDirectory="Release"
  ConfigurationType="2"
/>
  InheritedPropertySheets="$ (VCInstallDir)VCProjectDefaults\UpgradeFromVC71.vsprops"
  UseOfMFC="0"
  UseOfATL="0"
  ATLMinimizesCRunTimeLibraryUsage="false"
  CharSet="2"
  <Tool
    Name="VCPreBuildEventTool"
  />
  <Tool
    Name="VCCustomBuildTool"
  />
  <Tool
    Name="VCXMLDataGeneratorTool"
  />
  <Tool
    Name="VCWebServiceProxyGeneratorTool"
  />
  <Tool
    Name="VCMDLTool"
  />
  <Tool
    Name="VCCLCompilerTool"
    Optimization="2"
    AdditionalIncludeDirectories="..\tpccsapi\..\tpcc-c.ibm\include"
  />
  PreprocessorDefinitions="WIN32;NDEBUG;_WINDOWS;_USRDLL;TPCCDB2GLUE_EXPORTS"
  RuntimeLibrary="2"
  StructMemberAlignment="4"
  BufferSecurityCheck="false"
  DefaultCharSigned="true"
  UsePrecompiledHeader="0"
  WarningLevel="3"
  Detect64BitPortabilityProblems="true"
  DebugInformationFormat="3"
  CompileAs="2"
  />
</Tool>

```

```

    Name="VCManagedResourceCompilerTool"
  />
  <Tool
    Name="VCResourceCompilerTool"
  />
  <Tool
    Name="VCPreLinkEventTool"
  />
  <Tool
    Name="VCLinkerTool"
    AdditionalDependencies="..\tpc-c.ibm\Src.Cli\tpcccli.lib
c:\SQLLIB\lib\b2api.lib&quot; Advapi32.lib"
    OutputFile="$(OutDir)\tpccDB2glue.dll"
    LinkIncremental="1"
    AdditionalLibraryDirectories="&quot;C:\Program Files (x86)\Microsoft
Visual Studio 9.0\VC\lib&quot; &quot;C:\Program Files\Microsoft
SDKs\Windows\v6.0\ALib&quot;;C:\windows\system32"
    IgnoreAllDefaultLibraries="false"
    IgnoreDefaultLibraryNames="LIBCMT.lib"
    GenerateDebugInformation="false"
    SubSystem="2"
    OptimizeReferences="2"
    EnableCOMDATFolding="2"
    RandomizedBaseAddress="1"
    DataExecutionPrevention="0"
    ImportLibrary="$(OutDir)\tpccDB2glue.lib"
    TargetMachine="1"
  />
  <Tool
    Name="VCALinkTool"
  />
  <Tool
    Name="VCManifestTool"
    EmbedManifest="true"
  />
  <Tool
    Name="VCXDCMakeTool"
  />
  <Tool
    Name="VCBscMakeTool"
  />
  <Tool
    Name="VCFxCopTool"
  />
  <Tool
    Name="VCAppVerifierTool"
  />
  <Tool
    Name="VCPostBuildEventTool"
  />
</Configuration>
</Configurations>
<References>
</References>
<Files>
  <Filter
    Name="Source Files"
    Filter="*.cpp;c;*.cxx;*.def;*.odl;*.h;*.hpp;*.bat;*.asm;*.asmx"
    UniqueIdentifier="{4FC737F1-C7A5-4376-A066-2A32D752A2FF}"
  >
  <File
    RelativePath=".\stdafx.cpp"
  >
  <FileConfiguration
    Name="Debug|Win32"
  >
  <Tool
    Name="VCCLCompilerTool"
    UsePrecompiledHeader="1"
  />
  </FileConfiguration>
  </FileConfiguration>
  Name="Release|Win32"

```

```

  >
  <Tool
    Name="VCCLCompilerTool"
    UsePrecompiledHeader="1"
  />
  </FileConfiguration>
  </File>
  <File
    RelativePath=".\time.cpp"
  >
  </File>
  <File
    RelativePath=".\tpccDB2glue.cpp"
  >
  </File>
  </Filter>
  <Filter
    Name="Header Files"
    Filter="*.h;*.hpp;*.hxx;*.hm;*.inl;*.inc;*.xsd"
    UniqueIdentifier="{93995380-89BD-4b04-88EB-625FBE52EBFB}"
  >
  <File
    RelativePath=".\stdafx.h"
  >
  </File>
  <File
    RelativePath=".\tpccDB2glue.h"
  >
  </File>
  </Filter>
  <Filter
    Name="Resource Files"
    Filter="*.rc;*.ico;*.cur;*.bmp;*.dlg;*.rc2;*.rct;*.rgs;*.gif;*.jpg;*.jpeg;*.resx"
    UniqueIdentifier="{67DA6AB6-F800-4c08-8B7A-83BB121AAD01}"
  >
  </Filter>
  <File
    RelativePath=".\ReadMe.txt"
  >
  </File>
  <File
    RelativePath="..\tpccComReadMe.txt"
  >
  </File>
  </Files>
  <Globals>
  </Globals>
</VisualStudioProject>

```

# Appendix - B: Tunable Parameters

## B.1 Database Parameters.

### 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 = NO  
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) = 127325696  
Database memory threshold (DB\_MEM\_THRESH) = 10  
Max storage for lock list (4KB) (LOCKLIST) = 32000  
Percent. of lock lists per application (MAXLOCKS) = 20  
Package cache size (4KB) (PCKCACHE\_SZ) = 12000  
Sort heap thres for shared sorts (4KB) (SHEAPTHRES\_SHR) = 5000  
Sort list heap (4KB) (SORTHEAP) = 16

Database heap (4KB) (DBHEAP) = 524288  
Catalog cache size (4KB) (CATALOGCACHE\_SZ) = (MAXAPPLS\*5)  
Log buffer size (4KB) (LOGBUFSZ) = 60000  
Utilities heap size (4KB) (UTIL\_HEAP\_SZ) = 5000  
Buffer pool size (pages) (BUFFPAGE) = 1000

SQL statement heap (4KB) (STMTHAP) = 65000  
Default application heap (4KB) (APPLHEAPSZ) = 2500  
Application Memory Size (4KB) (APPL\_MEMORY) = 270000  
Statistics heap size (4KB) (STAT\_HEAP\_SZ) = AUTOMATIC(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\_IJCLEANERS) = 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) = 5050  
Average number of active applications (AVG\_APPLS) = 1  
Max DB files open per application (MAXFILOP) = 61440

Log file size (4KB) (LOGFILSIZ) = 262144  
Number of primary log files (LOGPRIMARY) = 250  
Number of secondary log files (LOGSECOND) = 0  
Changed path to log files (NEWLOGPATH) =  
Path to log files = /dev/rdbloglv  
Overflow log path (OVERFLOWLOGPATH) =  
Mirror log path (MIRRORLOGPATH) =  
First active log file = S0000001.LOG  
Block log on disk full (BLK\_LOG\_DSK\_FUL) = NO  
Block non logged operations (BLOCKNONLOGGED) = 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) = 3  
Percent log file reclaimed before soft ckcpt (SOFTMAX) = 7700  
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) = 3.070237e-07

Max number of concurrently active databases (NUMMDB) = 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/sqlib/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) = STAFF  
SYSCTRL group name (SYSCTRL\_GROUP) =  
SYSMAINT group name (SYSMAINT\_GROUP) =  
SYSMON group name (SYSMON\_GROUP) =

Client Userid-Password Plugin (CLNT\_PW\_PLUGIN) =  
Client Kerberos Plugin (CLNT\_KRB\_PLUGIN) =  
Group Plugin (GROUP\_PLUGIN) =  
GSS Plugin for Local Authorization (LOCAL\_GSSPLUGIN) =  
Server Plugin Mode (SRV\_PLUGIN\_MODE) = UNFENCED  
Server List of GSS Plugins (SRVCON\_GSSPLUGIN\_LIST) =  
Server Userid-Password Plugin (SRVCON\_PW\_PLUGIN) =  
Server Connection Authentication (SRVCON\_AUTH) = NOT\_SPECIFIED  
Cluster manager (CLUSTER\_MGR) =

Database manager authentication (AUTHENTICATION) = CLIENT  
Cataloging allowed without authority (CATALOG\_NOAUTH) = 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) = AUTOMATIC(90)  
 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) = 13128612  
 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) = 60  
 Agent pool size (NUM\_POOLAGENTS) = 0  
 Initial number of agents in pool (NUM\_INITAGENTS) = 0  
 Max number of coordinating agents (MAX\_COORDAGENTS) = AUTOMATIC(200)  
 Max number of client connections (MAX\_CONNECTIONS) = AUTOMATIC(MAX\_COORDAGENTS)

Keep fenced process (KEEPFENCED) = YES  
 Number of pooled fenced processes (FENCED\_POOL) = MAX\_COORDAGENTS  
 Initial number of fenced processes (NUM\_INITFENCED) = 0

Index re-creation time and redo index build (INDEXREC) = RESTART

Transaction manager database name (TM\_DATABASE) = 1ST\_CONN  
 Transaction resync interval (sec) (RESYNC\_INTERVAL) = 180

SPM name (SPM\_NAME) =  
 SPM log size (SPM\_LOG\_FILE\_SZ) = 256  
 SPM resync agent limit (SPM\_MAX\_RESYNC) = 20  
 SPM log path (SPM\_LOG\_PATH) =

TCP/IP Service name (SVCENAME) =  
 Discovery mode (DISCOVER) = SEARCH  
 Discover server instance (DISCOVER\_INST) = ENABLE

Maximum query degree of parallelism (MAX\_QUERYDEGREE) = ANY  
 Enable intra-partition parallelism (INTRA\_PARALLEL) = NO

No. of int. communication buffers(4KB)(FCM\_NUM\_BUFFERS) = AUTOMATIC(895)  
 No. of int. communication channels (FCM\_NUM\_CHANNELS) = AUTOMATIC(256)  
 db2start/db2stop timeout (min) (START\_STOP\_TIME) = 10

## db2set.cfg.out

[\*] DB2\_USE\_IOCP=YES  
 [\*] DB2\_KEEP\_AS\_AND\_DMS\_CONTAINERS\_OPEN=ON  
 [\*] DB2\_LARGE\_PAGE\_MEM=DB:16GB,APPL  
 [\*] DB2\_RESOURCE\_POLICY=/home/tpcc/tpc-c.ibm/cfg/affinity.cfg  
 [\*] DB2\_SELUDI\_COMM\_BUFFER=Y  
 [\*] DB2\_USE\_ALTERNATE\_PAGE\_CLEANSING=YES  
 [\*] DB2\_MAX\_NON\_TABLE\_LOCKS=1000  
 [\*] DB2\_TRUSTED\_BINDIN=ON  
 [\*] DB2\_KEEPTABLELOCK=CONNECTION  
 [\*] DB2\_NUM\_CKPW\_DAEMONS=0  
 [\*] DB2\_EVENT\_LOG\_CONFIG=OFF  
 [\*] DB2\_NO\_FORK\_CHECK=ON  
 [\*] DB2\_ALLOCATION\_SIZE=4194304  
 [\*] DB2\_APM\_PERFORMANCE=ALL  
 [\*] DB2\_PINNED\_BP=YES  
 [\*] DB2ASSUMEUPDATE=ON  
 [\*] DB2CHECKCLIENTINTERVAL=0  
 [\*] DB2\_HASH\_JOIN=OFF  
 [\*] DB2CHKSQLDA=OFF

[\*] DB2\_COLLECT\_TS\_REC\_INFO=false  
 [\*] DB2COMM=tcPIP  
 [\*] DB2CHKPTR=OFF

## affinity.cfg

```
<RESOURCE_POLICY>
<DATABASE_RESOURCE_POLICY>
<DBNAME>TPCC</DBNAME>
<METHOD>RSET</METHOD>
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00000</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtPcc0</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>1</NUM_CLEANERS>
<BUFFERPOOL_ID>4</BUFFERPOOL_ID>
<BUFFERPOOL_ID>12</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_ID>77</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00001</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtPcc1</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>1</NUM_CLEANERS>
<BUFFERPOOL_ID>5</BUFFERPOOL_ID>
<BUFFERPOOL_ID>13</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_ID>78</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
```

```
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00002</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtPcc2</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>1</NUM_CLEANERS>
<BUFFERPOOL_ID>6</BUFFERPOOL_ID>
<BUFFERPOOL_ID>14</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_ID>79</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
```

```
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00003</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtPcc3</SERVICE_NAME>
<BUFFERPOOL_BINDING>
```

```
<NUM_CLEANERS>1</NUM_CLEANERS>
<BUFFERPOOL_ID>7</BUFFERPOOL_ID>
<BUFFERPOOL_ID>15</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_ID>80</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
```

```
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00004</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtPcc4</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>1</NUM_CLEANERS>
<BUFFERPOOL_ID>8</BUFFERPOOL_ID>
<BUFFERPOOL_ID>16</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_ID>81</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
```

```
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00005</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtPcc5</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>1</NUM_CLEANERS>
<BUFFERPOOL_ID>9</BUFFERPOOL_ID>
<BUFFERPOOL_ID>17</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_ID>82</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
```

```
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00006</RESOURCE>
<DBMEM_PERCENTAGE>12.5</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtPcc6</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>1</NUM_CLEANERS>
<BUFFERPOOL_ID>10</BUFFERPOOL_ID>
<BUFFERPOOL_ID>18</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_ID>83</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
```

```
<RESOURCE_BINDING>
<RESOURCE>sys/node.05.00007</RESOURCE>
```





```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{D0264C87-0470-4A06-A0D7-63EF81968E35}]
"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,00,44,00,30,00,32,00,36,00,34,00,43,00,\
38,00,37,00,2d,00,30,00,34,00,37,00,30,00,2d,00,34,00,41,00,30,00,36,00,2d,\
00,41,00,30,00,44,00,37,00,2d,00,36,00,33,00,45,00,46,00,38,00,31,00,39,00,\
36,00,38,00,45,00,33,00,35,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{DF3B01C1-9D76-427F-BA7A-17AEAF182E30}]
"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,00,44,00,46,00,33,00,42,00,30,00,31,00,\
43,00,31,00,2d,00,39,00,44,00,37,00,36,00,2d,00,34,00,32,00,37,00,46,00,2d,\
00,42,00,41,00,37,00,41,00,2d,00,31,00,37,00,41,00,45,00,41,00,46,00,31,00,\
38,00,32,00,45,00,33,00,30,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{F7E95255-579D-4EF7-9A8B-20B4C40B7C48}]
"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,00,46,00,37,00,45,00,39,00,35,00,32,00,\
35,00,35,00,2d,00,35,00,37,00,39,00,44,00,2d,00,34,00,45,00,46,00,37,00,2d,\
00,39,00,41,00,38,00,42,00,2d,00,32,00,30,00,42,00,34,00,43,00,34,00,30,00,\
42,00,37,00,43,00,34,00,38,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DNSRegisteredAdapters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{02E71305-9C31-498F-9EC3-5AEA7EFEEB9}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000000
"LeaseObtainedTime"=dword:00000000
"T1"=dword:00000000
"T2"=dword:00000000
"LeaseTerminatesTime"=dword:00000000
"AddressType"=dword:00000000
"IsServerNapAware"=dword:00000000
"DhcpConnForceBroadcastFlag"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,36,00,38,00,2e,00,31,00,31,00,\
2e,00,31,00,30,00,31,00,00,00,00,00
"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
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{065f0c42-703a-11de-9954-806e6f6e6963}]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{1F5668B7-A26C-483F-B719-EF9338C9C4DC}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000000
"LeaseObtainedTime"=dword:00000000
```

```
"T1"=dword:00000000
"T2"=dword:00000000
"LeaseTerminatesTime"=dword:00000000
"AddressType"=dword:00000000
"IsServerNapAware"=dword:00000000
"DhcpConnForceBroadcastFlag"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,31,00,2e,00,32,00,00,00,00,\
00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{62C84429-9356-407B-9572-B798C6C8AC9A}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000000
"LeaseObtainedTime"=dword:00000000
"T1"=dword:00000000
"T2"=dword:00000000
"LeaseTerminatesTime"=dword:00000000
"AddressType"=dword:00000000
"IsServerNapAware"=dword:00000000
"DhcpConnForceBroadcastFlag"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,31,00,2e,00,32,00,00,00,00,\
00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{72E75E72-4FF7-4DE0-A81E-B4721EA039E8}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000000
"LeaseObtainedTime"=dword:00000000
"T1"=dword:00000000
"T2"=dword:00000000
"LeaseTerminatesTime"=dword:00000000
"AddressType"=dword:00000000
"IsServerNapAware"=dword:00000000
"DhcpConnForceBroadcastFlag"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,33,00,2e,00,31,00,2e,00,32,00,00,00,00,\
00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{D0264C87-0470-4A06-A0D7-63EF81968E35}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000000
"LeaseObtainedTime"=dword:00000000
"T1"=dword:00000000
"T2"=dword:00000000
"LeaseTerminatesTime"=dword:00000000
"AddressType"=dword:00000000
"IsServerNapAware"=dword:00000000
"DhcpConnForceBroadcastFlag"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,36,00,38,00,2e,00,31,00,33,00,\
2e,00,31,00,30,00,31,00,00,00,00,00
```

```
2e,00,31,00,30,00,30,00,00,00,00,00
"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
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{DF3B01C1-9D76-427F-BA7A-17AEAF182E30}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"NameServer"="9.0.7.1.9.0.6.11"
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000000
"LeaseObtainedTime"=dword:00000000
"T1"=dword:00000000
"T2"=dword:00000000
"LeaseTerminatesTime"=dword:00000000
"AddressType"=dword:00000000
"IsServerNapAware"=dword:00000000
"DhcpConnForceBroadcastFlag"=dword:00000000
"IPAddress"=hex(7):39,00,2e,00,33,00,2e,00,31,00,34,00,34,00,2e,00,31,00,37,00,\
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):39,00,2e,00,33,00,2e,00,31,00,34,00,34,00,2e,00,31,00,\
00,00,00,00
"DefaultGatewayMetric"=hex(7):30,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{F7E95255-579D-4EF7-9A8B-20B4C40B7C48}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000000
"LeaseObtainedTime"=dword:00000000
"T1"=dword:00000000
"T2"=dword:00000000
"LeaseTerminatesTime"=dword:00000000
"AddressType"=dword:00000000
"IsServerNapAware"=dword:00000000
"DhcpConnForceBroadcastFlag"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,32,00,2e,00,31,00,2e,00,32,00,00,00,\
00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes]
"0.0.0.0.0.0.0.0.9.3.144.1.-1"=""
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Winsock]
"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:08,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,02,00,\
00,00,02,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,00,00,00,00,02,00,\
00,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
03,00,00,00,00,00,00,00
"UseDelayedAcceptance"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Performance]
"Close"="CloseTcpipPerformanceData"
"Collect"="CollectTcpipPerformanceData"
```

```
"Library"=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,00,50,\
00,65,00,72,00,66,00,63,00,74,00,72,00,73,00,2e,00,64,00,6c,00,6c,00,00,00,\
"Open"="OpenTcplpPerformanceData"
"Object List"="502 510 546 548 582 638 658 1530 1532 1534"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcplp\ServiceProvider]
"Class"=dword:00000008
"DnsPriority"=dword:000007d0
"HostsPriority"=dword:000001f4
"LocalPriority"=dword:000001f3
"Name"="TCP/IP"
"NetbtPriority"=dword:000007d1
"ProviderPath"=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,\
77,00,73,00,6f,00,63,00,6b,00,33,00,32,00,2e,00,64,00,6c,00,6c,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcplp\Enum]
"0"="Root\LEGACY_TCPIP\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

## TPcc software registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\TPCC]
"dbName"="tpcc"
"dbPassword"="DB24aix"
"dbType"="DB2"
"dbUserName"="tpcc"
"divyLogPath"="c:\inetpub\wwwroot\tpcc\
"divyQueueLen"=dword:00004e20
"divyThreads"=dword:00000010
"errorLogFile"="c:\inetpub\wwwroot\tpcc\isapi_err.log"
"htmlTrace"=dword:00000000
"htmlTraceLogFile"="c:\inetpub\wwwroot\tpcc\isapi_log"
"isapi_trace"=dword:00000000
>nullDB"=dword:00000000
"numServers"=dword:00000001
"numUsers"=dword:0001e849
"numWarehouse"=dword:000030d4
```

## W3SVC registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,00,73,00,\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,73,00,76,00,63,00,68,00,6f,\
00,73,00,74,00,2e,00,65,00,78,00,65,00,20,00,2d,00,6b,00,20,00,69,00,69,00,\
73,00,73,00,76,00,63,00,73,00,00,00
"DisplayName"="@%windir%\system32\inetsrv\iisres.dll,-30003"
"DependOnService"=hex(7):57,00,41,00,53,00,00,00,48,00,54,00,54,00,50,00,00,00,\
00,00
"ObjectName"="LocalSystem"
"Description"="@%windir%\system32\inetsrv\iisres.dll,-30004"
"RequiredPrivileges"=hex(7):53,00,65,00,41,00,73,00,73,00,69,00,67,00,6e,00,50,\
00,72,00,69,00,6d,00,61,00,72,00,79,00,54,00,6f,00,6b,00,65,00,6e,00,50,00,\
72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,41,00,75,\
00,64,00,69,00,74,00,50,00,72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,\
00,00,53,00,65,00,42,00,61,00,63,00,6b,00,75,00,70,00,50,00,72,00,69,00,76,\
00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,43,00,68,00,61,00,6e,00,\
67,00,65,00,4e,00,6f,00,74,00,69,00,66,00,79,00,50,00,72,00,69,00,76,00,69,\
```

```
00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,43,00,72,00,65,00,61,00,74,00,\
65,00,47,00,6c,00,6f,00,62,00,61,00,6c,00,50,00,72,00,69,00,76,00,69,00,6c,\
00,65,00,67,00,65,00,00,00,53,00,65,00,44,00,65,00,62,00,75,00,67,00,50,00,\
72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,49,00,6d,\
00,70,00,65,00,72,00,73,00,6f,00,6e,00,61,00,74,00,65,00,50,00,72,00,69,00,\
76,00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,49,00,6e,00,63,00,72,\
00,65,00,61,00,73,00,65,00,51,00,75,00,6f,00,74,00,61,00,50,00,72,00,69,00,\
76,00,69,00,6c,00,65,00,67,00,65,00,00,00,53,00,65,00,52,00,65,00,73,00,74,\
00,6f,00,72,00,65,00,50,00,72,00,69,00,76,00,69,00,6c,00,65,00,67,00,65,00,\
00,00,53,00,65,00,54,00,63,00,62,00,50,00,72,00,69,00,76,00,69,00,6c,00,65,\
00,67,00,65,00,00,00,00,00
"FailureActions"=hex:ff,ff,ff,01,00,00,00,01,00,00,00,03,00,00,00,14,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000007
"MinorVersion"=dword:00000005
"AccessDeniedMessage"="Error: Access is Denied."
"InstallPath"=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,00,73,\
00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,74,00,\
73,00,72,00,76,00,00,00,00
"ServiceDll"=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,00,73,\
00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,74,00,\
73,00,72,00,76,00,5c,00,69,00,69,00,73,00,77,00,33,00,61,00,64,00,6d,00,2e,\
00,64,00,6c,00,6c,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\Advanc
edDataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSSe
rver.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]
"Close"="CloseW3PerformanceData"
"Open"="OpenW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Library"=hex(2):25,00,77,00,69,00,6e,00,64,00,69,00,72,00,25,00,5c,00,73,00,\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,74,00,73,\
00,72,00,76,00,5c,00,77,00,33,00,63,00,74,00,72,00,73,00,2e,00,64,00,6c,00,\
6c,00,00,00
"InstallType"=dword:00000001
"PerfInFile"="w3ctrs.in"
"First Counter"=dword:000015a2
"Last Counter"=dword:000016a8
"First Help"=dword:000015a3
"Last Help"=dword:000016a9
"Object List"="5538 5712"
```

## B.3 AIX Parameters

### IBM Power 780

#### Isattr -El sys0

SW_dist_intr	false	Enable SW distribution of interrupts	True
autorestart	true	Automatically REBOOT OS after a crash	True
boottype	disk	N/A	False
capacity_inc	1.00	Processor capacity increment	False
capped	true	Partition is capped	False
conslogin	enable	System Console Login	False
cpuguard	enable	CPU Guard	True
dedicated	true	Partition is dedicated	False
enhanced_RBAC	true	Enhanced RBAC Mode	True
ent_capacity	8.00	Entitled processor capacity	False
frequency	640000000	System Bus Frequency	False

fullcore	false	Enable full CORE dump	True
fwversion	IBM.ZM710_056	Firmware version and revision levels	False
ghostdev	0	Recreate devices in ODM on system change	True
id_to_partition	0x8000000000900001	Partition ID	False
id_to_system	0x8000000000900000	System ID	False
iostat	false	Continuously maintain DISK I/O history	True
keylock	normal	State of system keylock at boot time	False
log_pg_dealloc	true	Log predictive memory page deallocation events	True
max_capacity	8.00	Maximum potential processor capacity	False
max_logname	9	Maximum login name length at boot time	True
maxbuf	20	Maximum number of pages in block I/O BUFFER CACHE	True
maxmbuf	0	Maximum Kbytes of real memory allowed for MBUFS	True
maxpout	8193	HIGH water mark for pending write I/Os per file	True
maxuproc	100000	Maximum number of PROCESSES allowed per user	True
min_capacity	1.00	Minimum potential processor capacity	False
minpout	4096	LOW water mark for pending write I/Os per file	True
modelname	IBM,9117-MMB	Machine name	False
ncargs	256	ARG/ENV list size in 4K byte blocks	True
nfs4_acl_compat	secure	NFS4 ACL Compatibility Mode	True
pre430core	false	Use pre-430 style CORE dump	True
pre520tune	disable	Pre-520 tuning compatibility mode	True
realmem	528744448	Amount of usable physical memory in Kbytes	False
rtasversion	1	Open Firmware RTAS version	False
sed_config	select	Stack Execution Disable (SED) Mode	True
systemid	IBM,02SA00009	Hardware system identifier	False
variable_weight	0	Variable processor capacity weight	False

#### vmo -L

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE
DEPENDENCIES							
ame_cpus_per_pool	8	8	8	1	1K	processors	B
ame_maxfree_mem	0	24M	24M	192K	32G	bytes	D
ame_min_ucpool_size	0	0	0	5	95	% memory	D
ame_minfree_mem	0	8M	8M	64K	32767M	bytes	D
ams_loan_policy	n/a	1	1	0	2	numeric	D
force_realias_lite	0	0	0	0	1	boolean	D
kernel_heap_psize	16M	0	16M	0	16M	bytes	B
lgpg_regions	6832	0	6832	0	8E-1		D
lgpg_size	16M	0	16M	0	16M	bytes	D
low_ps_handling	1	1	1	1	2		D
maxfree	1088	1088	1088	16	24627K	4KB pages	D
minfree						memory_frames	
maxperm	2165K		2165K				S
maxpin	2972K		2972K				S
maxpin%	80	80	80	1	100	% memory	D
pinnable_frames						memory_frames	
memory_frames	126M		126M		4KB	pages	S
memplace_data	2	2	2	0	2		D
memplace_mapped_file	2	2	2	0	2		D

memplace_shm_anonymous	2	2	2	0	2		D
memplace_shm_named	2	2	2	0	2		D
memplace_stack	2	2	2	0	2		D
memplace_text	2	2	2	0	2		D
memplace_unmapped_file	2	2	2	0	2		D
minfree	960	960	960	8	24627K	4KB pages	D
maxfree							
memory_frames							
minperm	73927	73927					S
minperm%	3	3	3	1	100	% memory	D
nokilluid	0	0	0	0	4G-1	uid	D
npskill	36K	36K	36K	1	4M-1	4KB pages	D
npswam	144K	144K	144K	1	4M-1	4KB pages	D
numpsblks	4608K	4608K				4KB blocks	S
pinnable_frames	2346K	2346K				4KB pages	S
relalias_percentage	0	0	0	0	32K-1		D
scrub	0	0	0	0	1	boolean	D
v_pinshm	1	0	1	0	1	boolean	D
vmm_default_pspa	0	0	0	-1	100	numeric	D
wlm_memlimit_nonpg	1	1	1	0	1	boolean	D

n/a means parameter not supported by the current platform or kernel

Parameter types:  
S = Static: cannot be changed  
D = Dynamic: can be freely changed  
B = Bosboot: can only be changed using bosboot and reboot  
R = Reboot: can only be changed during reboot  
C = Connect: changes are only effective for future socket connections  
M = Mount: changes are only effective for future mountings  
I = Incremental: can only be incremented  
d = deprecated: deprecated and cannot be changed

Value conventions:  
K = Kilo: 2^10 G = Giga: 2^30 P = Peta: 2^50  
M = Mega: 2^20 T = Tera: 2^40 E = Exa: 2^60

### ioo -L

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE
DEPENDENCIES							
aio_active	1	1				boolean	S
aio_maxreqs	64K	64K	64K	4K	1M	numeric	D
aio_maxservers	30	30	30	1	20000	numeric	D
aio_minservers							
aio_minservers	3	3	3	0	20000	numeric	D
aio_maxservers							

aio_server_inactivity	300	300	300	1	86400	seconds	D
j2_atimeUpdateSymlink	0	0	0	0	1	boolean	D
j2_dynamicBufferPreallocation	16	16	16	0	256	16K slabs	D
j2_inodeCacheSize	10	400	10	1	1000		D
j2_maxPageReadAhead	128	128	128	0	64K	4KB pages	D
j2_maxRandomWrite	0	0	0	0	64K	4KB pages	D
j2_metadataCacheSize	10	400	10	1	1000		D
j2_minPageReadAhead	2	2	2	0	64K	4KB pages	D
j2_nPagesPerWriteBehindCluster	32	32	32	0	64K		D
j2_nRandomCluster	0	0	0	0	64K	16KB clusters	D
j2_syncPageCount	0	0	0	0	64K	4KB pages	D
j2_syncPageLimit	16	16	16	1	64K	iterations	D
lvm_bufont	9	9	9	1	64	128KB/buffer	D
maxpgahead	8	8	8	0	4K	4KB pages	D
maxrandwrt	0	0	0	0	2G-1	4KB pages	D
numclust	1	1	1	0	2G-1	16KB/cluster	D
numfsbufs	196	196	196	1	2G-1		M
pd_npages	64K	64K	64K	1	512K	4KB pages	D
posix_aio_active	0	0				boolean	S
posix_aio_maxreqs	64K	64K	64K	4K	1M	numeric	D
posix_aio_maxservers	30	30	30	1	20000	numeric	D
aio_minservers							
posix_aio_minservers	3	3	3	0	20000	numeric	D
aio_maxservers							
posix_aio_server_inactivity	300	300	300	1	86400	seconds	D

n/a means parameter not supported by the current platform or kernel

Parameter types:  
S = Static: cannot be changed  
D = Dynamic: can be freely changed  
B = Bosboot: can only be changed using bosboot and reboot  
R = Reboot: can only be changed during reboot  
C = Connect: changes are only effective for future socket connections  
M = Mount: changes are only effective for future mountings  
I = Incremental: can only be incremented  
d = deprecated: deprecated and cannot be changed

Value conventions:  
K = Kilo: 2^10 G = Giga: 2^30 P = Peta: 2^50  
M = Mega: 2^20 T = Tera: 2^40 E = Exa: 2^60



































```

ALTER TABLE CUSTOMER94 ADD CONSTRAINT CUSTOMER94CKC CHECK
(C_W_ID BETWEEN 80632 AND 81498);
SET INTEGRITY FOR CUSTOMER94 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER95 OFF;
ALTER TABLE CUSTOMER95 DROP CONSTRAINT CUSTOMER95CKC;
ALTER TABLE CUSTOMER95 ADD CONSTRAINT CUSTOMER95CKC CHECK
(C_W_ID BETWEEN 81499 AND 82365);
SET INTEGRITY FOR CUSTOMER95 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER96 OFF;
ALTER TABLE CUSTOMER96 DROP CONSTRAINT CUSTOMER96CKC;
ALTER TABLE CUSTOMER96 ADD CONSTRAINT CUSTOMER96CKC CHECK
(C_W_ID BETWEEN 82366 AND 83232);
SET INTEGRITY FOR CUSTOMER96 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER97 OFF;
ALTER TABLE CUSTOMER97 DROP CONSTRAINT CUSTOMER97CKC;
ALTER TABLE CUSTOMER97 ADD CONSTRAINT CUSTOMER97CKC CHECK
(C_W_ID BETWEEN 83233 AND 84099);
SET INTEGRITY FOR CUSTOMER97 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER98 OFF;
ALTER TABLE CUSTOMER98 DROP CONSTRAINT CUSTOMER98CKC;
ALTER TABLE CUSTOMER98 ADD CONSTRAINT CUSTOMER98CKC CHECK
(C_W_ID BETWEEN 84100 AND 84966);
SET INTEGRITY FOR CUSTOMER98 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER99 OFF;
ALTER TABLE CUSTOMER99 DROP CONSTRAINT CUSTOMER99CKC;
ALTER TABLE CUSTOMER99 ADD CONSTRAINT CUSTOMER99CKC CHECK
(C_W_ID BETWEEN 84967 AND 85833);
SET INTEGRITY FOR CUSTOMER99 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER100 OFF;
ALTER TABLE CUSTOMER100 DROP CONSTRAINT CUSTOMER100CKC;
ALTER TABLE CUSTOMER100 ADD CONSTRAINT CUSTOMER100CKC CHECK
(C_W_ID BETWEEN 85834 AND 86700);
SET INTEGRITY FOR CUSTOMER100 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER101 OFF;
ALTER TABLE CUSTOMER101 DROP CONSTRAINT CUSTOMER101CKC;
ALTER TABLE CUSTOMER101 ADD CONSTRAINT CUSTOMER101CKC CHECK
(C_W_ID BETWEEN 86701 AND 87567);
SET INTEGRITY FOR CUSTOMER101 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER102 OFF;
ALTER TABLE CUSTOMER102 DROP CONSTRAINT CUSTOMER102CKC;
ALTER TABLE CUSTOMER102 ADD CONSTRAINT CUSTOMER102CKC CHECK
(C_W_ID BETWEEN 87568 AND 88434);
SET INTEGRITY FOR CUSTOMER102 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER103 OFF;
ALTER TABLE CUSTOMER103 DROP CONSTRAINT CUSTOMER103CKC;
ALTER TABLE CUSTOMER103 ADD CONSTRAINT CUSTOMER103CKC CHECK
(C_W_ID BETWEEN 88435 AND 89301);
SET INTEGRITY FOR CUSTOMER103 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER104 OFF;
ALTER TABLE CUSTOMER104 DROP CONSTRAINT CUSTOMER104CKC;
ALTER TABLE CUSTOMER104 ADD CONSTRAINT CUSTOMER104CKC CHECK
(C_W_ID BETWEEN 89302 AND 90168);
SET INTEGRITY FOR CUSTOMER104 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER105 OFF;
ALTER TABLE CUSTOMER105 DROP CONSTRAINT CUSTOMER105CKC;
ALTER TABLE CUSTOMER105 ADD CONSTRAINT CUSTOMER105CKC CHECK
(C_W_ID BETWEEN 90169 AND 91035);
SET INTEGRITY FOR CUSTOMER105 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER106 OFF;
ALTER TABLE CUSTOMER106 DROP CONSTRAINT CUSTOMER106CKC;
ALTER TABLE CUSTOMER106 ADD CONSTRAINT CUSTOMER106CKC CHECK
(C_W_ID BETWEEN 91036 AND 91902);
SET INTEGRITY FOR CUSTOMER106 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER107 OFF;
ALTER TABLE CUSTOMER107 DROP CONSTRAINT CUSTOMER107CKC;
ALTER TABLE CUSTOMER107 ADD CONSTRAINT CUSTOMER107CKC CHECK
(C_W_ID BETWEEN 91903 AND 92769);
SET INTEGRITY FOR CUSTOMER107 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER108 OFF;
ALTER TABLE CUSTOMER108 DROP CONSTRAINT CUSTOMER108CKC;
ALTER TABLE CUSTOMER108 ADD CONSTRAINT CUSTOMER108CKC CHECK
(C_W_ID BETWEEN 92770 AND 93636);
SET INTEGRITY FOR CUSTOMER108 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER109 OFF;
ALTER TABLE CUSTOMER109 DROP CONSTRAINT CUSTOMER109CKC;
ALTER TABLE CUSTOMER109 ADD CONSTRAINT CUSTOMER109CKC CHECK
(C_W_ID BETWEEN 93637 AND 94503);
SET INTEGRITY FOR CUSTOMER109 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER110 OFF;
ALTER TABLE CUSTOMER110 DROP CONSTRAINT CUSTOMER110CKC;
ALTER TABLE CUSTOMER110 ADD CONSTRAINT CUSTOMER110CKC CHECK
(C_W_ID BETWEEN 94504 AND 95370);
SET INTEGRITY FOR CUSTOMER110 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER111 OFF;
ALTER TABLE CUSTOMER111 DROP CONSTRAINT CUSTOMER111CKC;
ALTER TABLE CUSTOMER111 ADD CONSTRAINT CUSTOMER111CKC CHECK
(C_W_ID BETWEEN 95371 AND 96237);
SET INTEGRITY FOR CUSTOMER111 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER112 OFF;
ALTER TABLE CUSTOMER112 DROP CONSTRAINT CUSTOMER112CKC;
ALTER TABLE CUSTOMER112 ADD CONSTRAINT CUSTOMER112CKC CHECK
(C_W_ID BETWEEN 96238 AND 97104);
SET INTEGRITY FOR CUSTOMER112 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER113 OFF;
ALTER TABLE CUSTOMER113 DROP CONSTRAINT CUSTOMER113CKC;
ALTER TABLE CUSTOMER113 ADD CONSTRAINT CUSTOMER113CKC CHECK
(C_W_ID BETWEEN 97105 AND 97971);
SET INTEGRITY FOR CUSTOMER113 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER114 OFF;
ALTER TABLE CUSTOMER114 DROP CONSTRAINT CUSTOMER114CKC;
ALTER TABLE CUSTOMER114 ADD CONSTRAINT CUSTOMER114CKC CHECK
(C_W_ID BETWEEN 97972 AND 98838);
SET INTEGRITY FOR CUSTOMER114 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER115 OFF;
ALTER TABLE CUSTOMER115 DROP CONSTRAINT CUSTOMER115CKC;

```

```

ALTER TABLE CUSTOMER115 ADD CONSTRAINT CUSTOMER115CKC CHECK
(C_W_ID BETWEEN 98839 AND 99705);
SET INTEGRITY FOR CUSTOMER115 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER116 OFF;
ALTER TABLE CUSTOMER116 DROP CONSTRAINT CUSTOMER116CKC;
ALTER TABLE CUSTOMER116 ADD CONSTRAINT CUSTOMER116CKC CHECK
(C_W_ID BETWEEN 99706 AND 100572);
SET INTEGRITY FOR CUSTOMER116 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER117 OFF;
ALTER TABLE CUSTOMER117 DROP CONSTRAINT CUSTOMER117CKC;
ALTER TABLE CUSTOMER117 ADD CONSTRAINT CUSTOMER117CKC CHECK
(C_W_ID BETWEEN 100573 AND 101439);
SET INTEGRITY FOR CUSTOMER117 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER118 OFF;
ALTER TABLE CUSTOMER118 DROP CONSTRAINT CUSTOMER118CKC;
ALTER TABLE CUSTOMER118 ADD CONSTRAINT CUSTOMER118CKC CHECK
(C_W_ID BETWEEN 101440 AND 102306);
SET INTEGRITY FOR CUSTOMER118 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER119 OFF;
ALTER TABLE CUSTOMER119 DROP CONSTRAINT CUSTOMER119CKC;
ALTER TABLE CUSTOMER119 ADD CONSTRAINT CUSTOMER119CKC CHECK
(C_W_ID BETWEEN 102307 AND 103173);
SET INTEGRITY FOR CUSTOMER119 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER120 OFF;
ALTER TABLE CUSTOMER120 DROP CONSTRAINT CUSTOMER120CKC;
ALTER TABLE CUSTOMER120 ADD CONSTRAINT CUSTOMER120CKC CHECK
(C_W_ID >= 103174);
SET INTEGRITY FOR CUSTOMER120 ALL IMMEDIATE UNCHECKED;
connect reset;

```

## DDL/CRCONST DISTRICT.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT1 OFF;
ALTER TABLE DISTRICT1 DROP CONSTRAINT DISTRICT1CKC;
ALTER TABLE DISTRICT1 ADD CONSTRAINT DISTRICT1CKC CHECK (D_W_ID
BETWEEN 1 AND 867);
SET INTEGRITY FOR DISTRICT1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT2 OFF;
ALTER TABLE DISTRICT2 DROP CONSTRAINT DISTRICT2CKC;
ALTER TABLE DISTRICT2 ADD CONSTRAINT DISTRICT2CKC CHECK (D_W_ID
BETWEEN 868 AND 1734);
SET INTEGRITY FOR DISTRICT2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT3 OFF;
ALTER TABLE DISTRICT3 DROP CONSTRAINT DISTRICT3CKC;
ALTER TABLE DISTRICT3 ADD CONSTRAINT DISTRICT3CKC CHECK (D_W_ID
BETWEEN 1735 AND 2601);
SET INTEGRITY FOR DISTRICT3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT4 OFF;
ALTER TABLE DISTRICT4 DROP CONSTRAINT DISTRICT4CKC;
ALTER TABLE DISTRICT4 ADD CONSTRAINT DISTRICT4CKC CHECK (D_W_ID
BETWEEN 2602 AND 3468);
SET INTEGRITY FOR DISTRICT4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT5 OFF;

```







```

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT100 OFF;
ALTER TABLE DISTRICT100 DROP CONSTRAINT DISTRICT100CKC;
ALTER TABLE DISTRICT100 ADD CONSTRAINT DISTRICT100CKC CHECK
(D_W_ID BETWEEN 85834 AND 86700);
SET INTEGRITY FOR DISTRICT100 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT101 OFF;
ALTER TABLE DISTRICT101 DROP CONSTRAINT DISTRICT101CKC;
ALTER TABLE DISTRICT101 ADD CONSTRAINT DISTRICT101CKC CHECK
(D_W_ID BETWEEN 86701 AND 87567);
SET INTEGRITY FOR DISTRICT101 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT102 OFF;
ALTER TABLE DISTRICT102 DROP CONSTRAINT DISTRICT102CKC;
ALTER TABLE DISTRICT102 ADD CONSTRAINT DISTRICT102CKC CHECK
(D_W_ID BETWEEN 87568 AND 88434);
SET INTEGRITY FOR DISTRICT102 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT103 OFF;
ALTER TABLE DISTRICT103 DROP CONSTRAINT DISTRICT103CKC;
ALTER TABLE DISTRICT103 ADD CONSTRAINT DISTRICT103CKC CHECK
(D_W_ID BETWEEN 88435 AND 89301);
SET INTEGRITY FOR DISTRICT103 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT104 OFF;
ALTER TABLE DISTRICT104 DROP CONSTRAINT DISTRICT104CKC;
ALTER TABLE DISTRICT104 ADD CONSTRAINT DISTRICT104CKC CHECK
(D_W_ID BETWEEN 89302 AND 90168);
SET INTEGRITY FOR DISTRICT104 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT105 OFF;
ALTER TABLE DISTRICT105 DROP CONSTRAINT DISTRICT105CKC;
ALTER TABLE DISTRICT105 ADD CONSTRAINT DISTRICT105CKC CHECK
(D_W_ID BETWEEN 90169 AND 91035);
SET INTEGRITY FOR DISTRICT105 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT106 OFF;
ALTER TABLE DISTRICT106 DROP CONSTRAINT DISTRICT106CKC;
ALTER TABLE DISTRICT106 ADD CONSTRAINT DISTRICT106CKC CHECK
(D_W_ID BETWEEN 91036 AND 91902);
SET INTEGRITY FOR DISTRICT106 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT107 OFF;
ALTER TABLE DISTRICT107 DROP CONSTRAINT DISTRICT107CKC;
ALTER TABLE DISTRICT107 ADD CONSTRAINT DISTRICT107CKC CHECK
(D_W_ID BETWEEN 91903 AND 92769);
SET INTEGRITY FOR DISTRICT107 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT108 OFF;
ALTER TABLE DISTRICT108 DROP CONSTRAINT DISTRICT108CKC;
ALTER TABLE DISTRICT108 ADD CONSTRAINT DISTRICT108CKC CHECK
(D_W_ID BETWEEN 92770 AND 93636);
SET INTEGRITY FOR DISTRICT108 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT109 OFF;
ALTER TABLE DISTRICT109 DROP CONSTRAINT DISTRICT109CKC;
ALTER TABLE DISTRICT109 ADD CONSTRAINT DISTRICT109CKC CHECK
(D_W_ID BETWEEN 93637 AND 94503);
SET INTEGRITY FOR DISTRICT109 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT110 OFF;
ALTER TABLE DISTRICT110 DROP CONSTRAINT DISTRICT110CKC;

```

```

ALTER TABLE DISTRICT110 ADD CONSTRAINT DISTRICT110CKC CHECK
(D_W_ID BETWEEN 94504 AND 95370);
SET INTEGRITY FOR DISTRICT110 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT111 OFF;
ALTER TABLE DISTRICT111 DROP CONSTRAINT DISTRICT111CKC;
ALTER TABLE DISTRICT111 ADD CONSTRAINT DISTRICT111CKC CHECK
(D_W_ID BETWEEN 95371 AND 96237);
SET INTEGRITY FOR DISTRICT111 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT112 OFF;
ALTER TABLE DISTRICT112 DROP CONSTRAINT DISTRICT112CKC;
ALTER TABLE DISTRICT112 ADD CONSTRAINT DISTRICT112CKC CHECK
(D_W_ID BETWEEN 96238 AND 97104);
SET INTEGRITY FOR DISTRICT112 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT113 OFF;
ALTER TABLE DISTRICT113 DROP CONSTRAINT DISTRICT113CKC;
ALTER TABLE DISTRICT113 ADD CONSTRAINT DISTRICT113CKC CHECK
(D_W_ID BETWEEN 97105 AND 97971);
SET INTEGRITY FOR DISTRICT113 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT114 OFF;
ALTER TABLE DISTRICT114 DROP CONSTRAINT DISTRICT114CKC;
ALTER TABLE DISTRICT114 ADD CONSTRAINT DISTRICT114CKC CHECK
(D_W_ID BETWEEN 97972 AND 98838);
SET INTEGRITY FOR DISTRICT114 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT115 OFF;
ALTER TABLE DISTRICT115 DROP CONSTRAINT DISTRICT115CKC;
ALTER TABLE DISTRICT115 ADD CONSTRAINT DISTRICT115CKC CHECK
(D_W_ID BETWEEN 98839 AND 99705);
SET INTEGRITY FOR DISTRICT115 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT116 OFF;
ALTER TABLE DISTRICT116 DROP CONSTRAINT DISTRICT116CKC;
ALTER TABLE DISTRICT116 ADD CONSTRAINT DISTRICT116CKC CHECK
(D_W_ID BETWEEN 99706 AND 100572);
SET INTEGRITY FOR DISTRICT116 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT117 OFF;
ALTER TABLE DISTRICT117 DROP CONSTRAINT DISTRICT117CKC;
ALTER TABLE DISTRICT117 ADD CONSTRAINT DISTRICT117CKC CHECK
(D_W_ID BETWEEN 100573 AND 101439);
SET INTEGRITY FOR DISTRICT117 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT118 OFF;
ALTER TABLE DISTRICT118 DROP CONSTRAINT DISTRICT118CKC;
ALTER TABLE DISTRICT118 ADD CONSTRAINT DISTRICT118CKC CHECK
(D_W_ID BETWEEN 101440 AND 102306);
SET INTEGRITY FOR DISTRICT118 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT119 OFF;
ALTER TABLE DISTRICT119 DROP CONSTRAINT DISTRICT119CKC;
ALTER TABLE DISTRICT119 ADD CONSTRAINT DISTRICT119CKC CHECK
(D_W_ID BETWEEN 102307 AND 103173);
SET INTEGRITY FOR DISTRICT119 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT120 OFF;
ALTER TABLE DISTRICT120 DROP CONSTRAINT DISTRICT120CKC;
ALTER TABLE DISTRICT120 ADD CONSTRAINT DISTRICT120CKC CHECK
(D_W_ID >= 103174);
SET INTEGRITY FOR DISTRICT120 ALL IMMEDIATE UNCHECKED;
connect reset;

```

## DDL/CRCONST\_HISTORY.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY1 DROP CONSTRAINT HISTORY1CKC;
ALTER TABLE HISTORY1 ADD CONSTRAINT HISTORY1CKC CHECK (H_W_ID
BETWEEN 1 AND 867);
SET INTEGRITY FOR HISTORY1 ALL IMMEDIATE UNCHECKED;
connect reset;
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 868 AND 1734);
SET INTEGRITY FOR HISTORY2 ALL IMMEDIATE UNCHECKED;
connect reset;
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 1735 AND 2601);
SET INTEGRITY FOR HISTORY3 ALL IMMEDIATE UNCHECKED;
connect reset;
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 2602 AND 3468);
SET INTEGRITY FOR HISTORY4 ALL IMMEDIATE UNCHECKED;
connect reset;
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 3469 AND 4335);
SET INTEGRITY FOR HISTORY5 ALL IMMEDIATE UNCHECKED;
connect reset;
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 4336 AND 5202);
SET INTEGRITY FOR HISTORY6 ALL IMMEDIATE UNCHECKED;
connect reset;
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 5203 AND 6069);
SET INTEGRITY FOR HISTORY7 ALL IMMEDIATE UNCHECKED;
connect reset;
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 6070 AND 6936);
SET INTEGRITY FOR HISTORY8 ALL IMMEDIATE UNCHECKED;
connect reset;
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 6937 AND 7803);
SET INTEGRITY FOR HISTORY9 ALL IMMEDIATE UNCHECKED;
connect reset;
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 7804 AND 8670);
SET INTEGRITY FOR HISTORY10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;

```









```

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY106 OFF;
ALTER TABLE HISTORY106 DROP CONSTRAINT HISTORY106CKC;
ALTER TABLE HISTORY106 ADD CONSTRAINT HISTORY106CKC CHECK
(H_W_ID BETWEEN 91036 AND 91902);
SET INTEGRITY FOR HISTORY106 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY107 OFF;
ALTER TABLE HISTORY107 DROP CONSTRAINT HISTORY107CKC;
ALTER TABLE HISTORY107 ADD CONSTRAINT HISTORY107CKC CHECK
(H_W_ID BETWEEN 91903 AND 92769);
SET INTEGRITY FOR HISTORY107 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY108 OFF;
ALTER TABLE HISTORY108 DROP CONSTRAINT HISTORY108CKC;
ALTER TABLE HISTORY108 ADD CONSTRAINT HISTORY108CKC CHECK
(H_W_ID BETWEEN 92770 AND 93636);
SET INTEGRITY FOR HISTORY108 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY109 OFF;
ALTER TABLE HISTORY109 DROP CONSTRAINT HISTORY109CKC;
ALTER TABLE HISTORY109 ADD CONSTRAINT HISTORY109CKC CHECK
(H_W_ID BETWEEN 93637 AND 94503);
SET INTEGRITY FOR HISTORY109 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY110 OFF;
ALTER TABLE HISTORY110 DROP CONSTRAINT HISTORY110CKC;
ALTER TABLE HISTORY110 ADD CONSTRAINT HISTORY110CKC CHECK
(H_W_ID BETWEEN 94504 AND 95370);
SET INTEGRITY FOR HISTORY110 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY111 OFF;
ALTER TABLE HISTORY111 DROP CONSTRAINT HISTORY111CKC;
ALTER TABLE HISTORY111 ADD CONSTRAINT HISTORY111CKC CHECK
(H_W_ID BETWEEN 95371 AND 96237);
SET INTEGRITY FOR HISTORY111 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY112 OFF;
ALTER TABLE HISTORY112 DROP CONSTRAINT HISTORY112CKC;
ALTER TABLE HISTORY112 ADD CONSTRAINT HISTORY112CKC CHECK
(H_W_ID BETWEEN 96238 AND 97104);
SET INTEGRITY FOR HISTORY112 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY113 OFF;
ALTER TABLE HISTORY113 DROP CONSTRAINT HISTORY113CKC;
ALTER TABLE HISTORY113 ADD CONSTRAINT HISTORY113CKC CHECK
(H_W_ID BETWEEN 97105 AND 97971);
SET INTEGRITY FOR HISTORY113 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY114 OFF;
ALTER TABLE HISTORY114 DROP CONSTRAINT HISTORY114CKC;
ALTER TABLE HISTORY114 ADD CONSTRAINT HISTORY114CKC CHECK
(H_W_ID BETWEEN 97972 AND 98838);
SET INTEGRITY FOR HISTORY114 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY115 OFF;
ALTER TABLE HISTORY115 DROP CONSTRAINT HISTORY115CKC;
ALTER TABLE HISTORY115 ADD CONSTRAINT HISTORY115CKC CHECK
(H_W_ID BETWEEN 98839 AND 99705);
SET INTEGRITY FOR HISTORY115 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY116 OFF;
ALTER TABLE HISTORY116 DROP CONSTRAINT HISTORY116CKC;

```

```

ALTER TABLE HISTORY116 ADD CONSTRAINT HISTORY116CKC CHECK
(H_W_ID BETWEEN 99706 AND 100572);
SET INTEGRITY FOR HISTORY116 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY117 OFF;
ALTER TABLE HISTORY117 DROP CONSTRAINT HISTORY117CKC;
ALTER TABLE HISTORY117 ADD CONSTRAINT HISTORY117CKC CHECK
(H_W_ID BETWEEN 100573 AND 101439);
SET INTEGRITY FOR HISTORY117 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY118 OFF;
ALTER TABLE HISTORY118 DROP CONSTRAINT HISTORY118CKC;
ALTER TABLE HISTORY118 ADD CONSTRAINT HISTORY118CKC CHECK
(H_W_ID BETWEEN 101440 AND 102306);
SET INTEGRITY FOR HISTORY118 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY119 OFF;
ALTER TABLE HISTORY119 DROP CONSTRAINT HISTORY119CKC;
ALTER TABLE HISTORY119 ADD CONSTRAINT HISTORY119CKC CHECK
(H_W_ID BETWEEN 102307 AND 103173);
SET INTEGRITY FOR HISTORY119 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY120 OFF;
ALTER TABLE HISTORY120 DROP CONSTRAINT HISTORY120CKC;
ALTER TABLE HISTORY120 ADD CONSTRAINT HISTORY120CKC CHECK
(H_W_ID >= 103174);
SET INTEGRITY FOR HISTORY120 ALL IMMEDIATE UNCHECKED;
connect reset;

```

### **DDL/CRCONST NEW ORDERA.ddl**

```

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA1 OFF;
ALTER TABLE NEW_ORDERA1 DROP CONSTRAINT NEW_ORDERA1CKC;
ALTER TABLE NEW_ORDERA1 ADD CONSTRAINT NEW_ORDERA1CKC CHECK
((NO_W_ID BETWEEN 1 AND 867) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA2 OFF;
ALTER TABLE NEW_ORDERA2 DROP CONSTRAINT NEW_ORDERA2CKC;
ALTER TABLE NEW_ORDERA2 ADD CONSTRAINT NEW_ORDERA2CKC CHECK
((NO_W_ID BETWEEN 868 AND 1734) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA3 OFF;
ALTER TABLE NEW_ORDERA3 DROP CONSTRAINT NEW_ORDERA3CKC;
ALTER TABLE NEW_ORDERA3 ADD CONSTRAINT NEW_ORDERA3CKC CHECK
((NO_W_ID BETWEEN 1735 AND 2601) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA4 OFF;
ALTER TABLE NEW_ORDERA4 DROP CONSTRAINT NEW_ORDERA4CKC;
ALTER TABLE NEW_ORDERA4 ADD CONSTRAINT NEW_ORDERA4CKC CHECK
((NO_W_ID BETWEEN 2602 AND 3468) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA5 OFF;
ALTER TABLE NEW_ORDERA5 DROP CONSTRAINT NEW_ORDERA5CKC;
ALTER TABLE NEW_ORDERA5 ADD CONSTRAINT NEW_ORDERA5CKC CHECK
((NO_W_ID BETWEEN 3469 AND 4335) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA6 OFF;

```

```

ALTER TABLE NEW_ORDERA6 DROP CONSTRAINT NEW_ORDERA6CKC;
ALTER TABLE NEW_ORDERA6 ADD CONSTRAINT NEW_ORDERA6CKC CHECK
((NO_W_ID BETWEEN 4336 AND 5202) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA7 OFF;
ALTER TABLE NEW_ORDERA7 DROP CONSTRAINT NEW_ORDERA7CKC;
ALTER TABLE NEW_ORDERA7 ADD CONSTRAINT NEW_ORDERA7CKC CHECK
((NO_W_ID BETWEEN 5203 AND 6069) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA8 OFF;
ALTER TABLE NEW_ORDERA8 DROP CONSTRAINT NEW_ORDERA8CKC;
ALTER TABLE NEW_ORDERA8 ADD CONSTRAINT NEW_ORDERA8CKC CHECK
((NO_W_ID BETWEEN 6070 AND 6936) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA9 OFF;
ALTER TABLE NEW_ORDERA9 DROP CONSTRAINT NEW_ORDERA9CKC;
ALTER TABLE NEW_ORDERA9 ADD CONSTRAINT NEW_ORDERA9CKC CHECK
((NO_W_ID BETWEEN 6937 AND 7803) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA10 OFF;
ALTER TABLE NEW_ORDERA10 DROP CONSTRAINT NEW_ORDERA10CKC;
ALTER TABLE NEW_ORDERA10 ADD CONSTRAINT NEW_ORDERA10CKC CHECK
((NO_W_ID BETWEEN 7804 AND 8670) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA11 OFF;
ALTER TABLE NEW_ORDERA11 DROP CONSTRAINT NEW_ORDERA11CKC;
ALTER TABLE NEW_ORDERA11 ADD CONSTRAINT NEW_ORDERA11CKC CHECK
((NO_W_ID BETWEEN 8671 AND 9537) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA12 OFF;
ALTER TABLE NEW_ORDERA12 DROP CONSTRAINT NEW_ORDERA12CKC;
ALTER TABLE NEW_ORDERA12 ADD CONSTRAINT NEW_ORDERA12CKC CHECK
((NO_W_ID BETWEEN 9538 AND 10404) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA13 OFF;
ALTER TABLE NEW_ORDERA13 DROP CONSTRAINT NEW_ORDERA13CKC;
ALTER TABLE NEW_ORDERA13 ADD CONSTRAINT NEW_ORDERA13CKC CHECK
((NO_W_ID BETWEEN 10405 AND 11271) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA14 OFF;
ALTER TABLE NEW_ORDERA14 DROP CONSTRAINT NEW_ORDERA14CKC;
ALTER TABLE NEW_ORDERA14 ADD CONSTRAINT NEW_ORDERA14CKC CHECK
((NO_W_ID BETWEEN 11272 AND 12138) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA15 OFF;
ALTER TABLE NEW_ORDERA15 DROP CONSTRAINT NEW_ORDERA15CKC;
ALTER TABLE NEW_ORDERA15 ADD CONSTRAINT NEW_ORDERA15CKC CHECK
((NO_W_ID BETWEEN 12139 AND 13005) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERA16 OFF;
ALTER TABLE NEW_ORDERA16 DROP CONSTRAINT NEW_ORDERA16CKC;
ALTER TABLE NEW_ORDERA16 ADD CONSTRAINT NEW_ORDERA16CKC CHECK
((NO_W_ID BETWEEN 13006 AND 13872) AND (NO_O_ID <= 3741));
SET INTEGRITY FOR NEW_ORDERA16 ALL IMMEDIATE UNCHECKED;

```

















```

connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB117 OFF;
ALTER TABLE NEW_ORDERB117 DROP CONSTRAINT NEW_ORDERB117CKC;
ALTER TABLE NEW_ORDERB117 ADD CONSTRAINT NEW_ORDERB117CKC CHECK
((NO_W_ID BETWEEN 100573 AND 101439) AND (NO_O_ID >= 3742));
SET INTEGRITY FOR NEW_ORDERB117 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB118 OFF;
ALTER TABLE NEW_ORDERB118 DROP CONSTRAINT NEW_ORDERB118CKC;
ALTER TABLE NEW_ORDERB118 ADD CONSTRAINT NEW_ORDERB118CKC CHECK
((NO_W_ID BETWEEN 101440 AND 102306) AND (NO_O_ID >= 3742));
SET INTEGRITY FOR NEW_ORDERB118 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB119 OFF;
ALTER TABLE NEW_ORDERB119 DROP CONSTRAINT NEW_ORDERB119CKC;
ALTER TABLE NEW_ORDERB119 ADD CONSTRAINT NEW_ORDERB119CKC CHECK
((NO_W_ID BETWEEN 102307 AND 103173) AND (NO_O_ID >= 3742));
SET INTEGRITY FOR NEW_ORDERB119 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB120 OFF;
ALTER TABLE NEW_ORDERB120 DROP CONSTRAINT NEW_ORDERB120CKC;
ALTER TABLE NEW_ORDERB120 ADD CONSTRAINT NEW_ORDERB120CKC CHECK
((NO_W_ID >= 103174) AND (NO_O_ID >= 3742));
SET INTEGRITY FOR NEW_ORDERB120 ALL IMMEDIATE UNCHECKED;
connect reset;

```

## DDL/CRCONST ORDERS.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS1 OFF;
ALTER TABLE ORDERS1 DROP CONSTRAINT ORDERS1CKC;
ALTER TABLE ORDERS1 ADD CONSTRAINT ORDERS1CKC CHECK (O_W_ID
BETWEEN 1 AND 867);
SET INTEGRITY FOR ORDERS1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS2 OFF;
ALTER TABLE ORDERS2 DROP CONSTRAINT ORDERS2CKC;
ALTER TABLE ORDERS2 ADD CONSTRAINT ORDERS2CKC CHECK (O_W_ID
BETWEEN 868 AND 1734);
SET INTEGRITY FOR ORDERS2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS3 OFF;
ALTER TABLE ORDERS3 DROP CONSTRAINT ORDERS3CKC;
ALTER TABLE ORDERS3 ADD CONSTRAINT ORDERS3CKC CHECK (O_W_ID
BETWEEN 1735 AND 2601);
SET INTEGRITY FOR ORDERS3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS4 OFF;
ALTER TABLE ORDERS4 DROP CONSTRAINT ORDERS4CKC;
ALTER TABLE ORDERS4 ADD CONSTRAINT ORDERS4CKC CHECK (O_W_ID
BETWEEN 2602 AND 3468);
SET INTEGRITY FOR ORDERS4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS5 OFF;
ALTER TABLE ORDERS5 DROP CONSTRAINT ORDERS5CKC;
ALTER TABLE ORDERS5 ADD CONSTRAINT ORDERS5CKC CHECK (O_W_ID
BETWEEN 3469 AND 4335);
SET INTEGRITY FOR ORDERS5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS6 OFF;
ALTER TABLE ORDERS6 DROP CONSTRAINT ORDERS6CKC;
ALTER TABLE ORDERS6 ADD CONSTRAINT ORDERS6CKC CHECK (O_W_ID
BETWEEN 4336 AND 5202);
SET INTEGRITY FOR ORDERS6 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS7 OFF;
ALTER TABLE ORDERS7 DROP CONSTRAINT ORDERS7CKC;
ALTER TABLE ORDERS7 ADD CONSTRAINT ORDERS7CKC CHECK (O_W_ID
BETWEEN 5203 AND 6069);
SET INTEGRITY FOR ORDERS7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS8 OFF;
ALTER TABLE ORDERS8 DROP CONSTRAINT ORDERS8CKC;
ALTER TABLE ORDERS8 ADD CONSTRAINT ORDERS8CKC CHECK (O_W_ID
BETWEEN 6070 AND 6936);
SET INTEGRITY FOR ORDERS8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS9 OFF;
ALTER TABLE ORDERS9 DROP CONSTRAINT ORDERS9CKC;
ALTER TABLE ORDERS9 ADD CONSTRAINT ORDERS9CKC CHECK (O_W_ID
BETWEEN 6937 AND 7803);
SET INTEGRITY FOR ORDERS9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS10 OFF;
ALTER TABLE ORDERS10 DROP CONSTRAINT ORDERS10CKC;
ALTER TABLE ORDERS10 ADD CONSTRAINT ORDERS10CKC CHECK (O_W_ID
BETWEEN 7804 AND 8670);
SET INTEGRITY FOR ORDERS10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS11 OFF;
ALTER TABLE ORDERS11 DROP CONSTRAINT ORDERS11CKC;
ALTER TABLE ORDERS11 ADD CONSTRAINT ORDERS11CKC CHECK (O_W_ID
BETWEEN 8671 AND 9537);
SET INTEGRITY FOR ORDERS11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS12 OFF;
ALTER TABLE ORDERS12 DROP CONSTRAINT ORDERS12CKC;
ALTER TABLE ORDERS12 ADD CONSTRAINT ORDERS12CKC CHECK (O_W_ID
BETWEEN 9538 AND 10404);
SET INTEGRITY FOR ORDERS12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS13 OFF;
ALTER TABLE ORDERS13 DROP CONSTRAINT ORDERS13CKC;
ALTER TABLE ORDERS13 ADD CONSTRAINT ORDERS13CKC CHECK (O_W_ID
BETWEEN 10405 AND 11271);
SET INTEGRITY FOR ORDERS13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS14 OFF;
ALTER TABLE ORDERS14 DROP CONSTRAINT ORDERS14CKC;
ALTER TABLE ORDERS14 ADD CONSTRAINT ORDERS14CKC CHECK (O_W_ID
BETWEEN 11272 AND 12138);
SET INTEGRITY FOR ORDERS14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS15 OFF;
ALTER TABLE ORDERS15 DROP CONSTRAINT ORDERS15CKC;
ALTER TABLE ORDERS15 ADD CONSTRAINT ORDERS15CKC CHECK (O_W_ID
BETWEEN 12139 AND 13005);
SET INTEGRITY FOR ORDERS15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS16 OFF;
ALTER TABLE ORDERS16 DROP CONSTRAINT ORDERS16CKC;
ALTER TABLE ORDERS16 ADD CONSTRAINT ORDERS16CKC CHECK (O_W_ID
BETWEEN 13006 AND 13872);
SET INTEGRITY FOR ORDERS16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS17 OFF;
ALTER TABLE ORDERS17 DROP CONSTRAINT ORDERS17CKC;

```

```

ALTER TABLE ORDERS17 ADD CONSTRAINT ORDERS17CKC CHECK (O_W_ID
BETWEEN 13873 AND 14739);
SET INTEGRITY FOR ORDERS17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS18 OFF;
ALTER TABLE ORDERS18 DROP CONSTRAINT ORDERS18CKC;
ALTER TABLE ORDERS18 ADD CONSTRAINT ORDERS18CKC CHECK (O_W_ID
BETWEEN 14740 AND 15606);
SET INTEGRITY FOR ORDERS18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS19 OFF;
ALTER TABLE ORDERS19 DROP CONSTRAINT ORDERS19CKC;
ALTER TABLE ORDERS19 ADD CONSTRAINT ORDERS19CKC CHECK (O_W_ID
BETWEEN 15607 AND 16473);
SET INTEGRITY FOR ORDERS19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS20 OFF;
ALTER TABLE ORDERS20 DROP CONSTRAINT ORDERS20CKC;
ALTER TABLE ORDERS20 ADD CONSTRAINT ORDERS20CKC CHECK (O_W_ID
BETWEEN 16474 AND 17340);
SET INTEGRITY FOR ORDERS20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS21 OFF;
ALTER TABLE ORDERS21 DROP CONSTRAINT ORDERS21CKC;
ALTER TABLE ORDERS21 ADD CONSTRAINT ORDERS21CKC CHECK (O_W_ID
BETWEEN 17341 AND 18207);
SET INTEGRITY FOR ORDERS21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS22 OFF;
ALTER TABLE ORDERS22 DROP CONSTRAINT ORDERS22CKC;
ALTER TABLE ORDERS22 ADD CONSTRAINT ORDERS22CKC CHECK (O_W_ID
BETWEEN 18208 AND 19074);
SET INTEGRITY FOR ORDERS22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS23 OFF;
ALTER TABLE ORDERS23 DROP CONSTRAINT ORDERS23CKC;
ALTER TABLE ORDERS23 ADD CONSTRAINT ORDERS23CKC CHECK (O_W_ID
BETWEEN 19075 AND 19941);
SET INTEGRITY FOR ORDERS23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS24 OFF;
ALTER TABLE ORDERS24 DROP CONSTRAINT ORDERS24CKC;
ALTER TABLE ORDERS24 ADD CONSTRAINT ORDERS24CKC CHECK (O_W_ID
BETWEEN 19942 AND 20808);
SET INTEGRITY FOR ORDERS24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS25 OFF;
ALTER TABLE ORDERS25 DROP CONSTRAINT ORDERS25CKC;
ALTER TABLE ORDERS25 ADD CONSTRAINT ORDERS25CKC CHECK (O_W_ID
BETWEEN 20809 AND 21675);
SET INTEGRITY FOR ORDERS25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS26 OFF;
ALTER TABLE ORDERS26 DROP CONSTRAINT ORDERS26CKC;
ALTER TABLE ORDERS26 ADD CONSTRAINT ORDERS26CKC CHECK (O_W_ID
BETWEEN 21676 AND 22542);
SET INTEGRITY FOR ORDERS26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS27 OFF;
ALTER TABLE ORDERS27 DROP CONSTRAINT ORDERS27CKC;
ALTER TABLE ORDERS27 ADD CONSTRAINT ORDERS27CKC CHECK (O_W_ID
BETWEEN 22543 AND 23409);
SET INTEGRITY FOR ORDERS27 ALL IMMEDIATE UNCHECKED;
connect reset;

```

















```

ALTER TABLE STOCK7 DROP CONSTRAINT STOCK7CKC;
ALTER TABLE STOCK7 ADD CONSTRAINT STOCK7CKC CHECK (S_W_ID
BETWEEN 5203 AND 6069);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK8 OFF;
ALTER TABLE STOCK8 DROP CONSTRAINT STOCK8CKC;
ALTER TABLE STOCK8 ADD CONSTRAINT STOCK8CKC CHECK (S_W_ID
BETWEEN 6070 AND 6936);
SET INTEGRITY FOR STOCK8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK9 OFF;
ALTER TABLE STOCK9 DROP CONSTRAINT STOCK9CKC;
ALTER TABLE STOCK9 ADD CONSTRAINT STOCK9CKC CHECK (S_W_ID
BETWEEN 6937 AND 7803);
SET INTEGRITY FOR STOCK9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK10 OFF;
ALTER TABLE STOCK10 DROP CONSTRAINT STOCK10CKC;
ALTER TABLE STOCK10 ADD CONSTRAINT STOCK10CKC CHECK (S_W_ID
BETWEEN 7804 AND 8670);
SET INTEGRITY FOR STOCK10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK11 OFF;
ALTER TABLE STOCK11 DROP CONSTRAINT STOCK11CKC;
ALTER TABLE STOCK11 ADD CONSTRAINT STOCK11CKC CHECK (S_W_ID
BETWEEN 8671 AND 9537);
SET INTEGRITY FOR STOCK11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK12 OFF;
ALTER TABLE STOCK12 DROP CONSTRAINT STOCK12CKC;
ALTER TABLE STOCK12 ADD CONSTRAINT STOCK12CKC CHECK (S_W_ID
BETWEEN 9538 AND 10404);
SET INTEGRITY FOR STOCK12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK13 OFF;
ALTER TABLE STOCK13 DROP CONSTRAINT STOCK13CKC;
ALTER TABLE STOCK13 ADD CONSTRAINT STOCK13CKC CHECK (S_W_ID
BETWEEN 10405 AND 11271);
SET INTEGRITY FOR STOCK13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK14 OFF;
ALTER TABLE STOCK14 DROP CONSTRAINT STOCK14CKC;
ALTER TABLE STOCK14 ADD CONSTRAINT STOCK14CKC CHECK (S_W_ID
BETWEEN 11272 AND 12138);
SET INTEGRITY FOR STOCK14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK15 OFF;
ALTER TABLE STOCK15 DROP CONSTRAINT STOCK15CKC;
ALTER TABLE STOCK15 ADD CONSTRAINT STOCK15CKC CHECK (S_W_ID
BETWEEN 12139 AND 13005);
SET INTEGRITY FOR STOCK15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK16 OFF;
ALTER TABLE STOCK16 DROP CONSTRAINT STOCK16CKC;
ALTER TABLE STOCK16 ADD CONSTRAINT STOCK16CKC CHECK (S_W_ID
BETWEEN 13006 AND 13872);
SET INTEGRITY FOR STOCK16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK17 OFF;
ALTER TABLE STOCK17 DROP CONSTRAINT STOCK17CKC;
ALTER TABLE STOCK17 ADD CONSTRAINT STOCK17CKC CHECK (S_W_ID
BETWEEN 13873 AND 14739);
SET INTEGRITY FOR STOCK17 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK18 OFF;
ALTER TABLE STOCK18 DROP CONSTRAINT STOCK18CKC;
ALTER TABLE STOCK18 ADD CONSTRAINT STOCK18CKC CHECK (S_W_ID
BETWEEN 14740 AND 15606);
SET INTEGRITY FOR STOCK18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK19 OFF;
ALTER TABLE STOCK19 DROP CONSTRAINT STOCK19CKC;
ALTER TABLE STOCK19 ADD CONSTRAINT STOCK19CKC CHECK (S_W_ID
BETWEEN 15607 AND 16473);
SET INTEGRITY FOR STOCK19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK20 OFF;
ALTER TABLE STOCK20 DROP CONSTRAINT STOCK20CKC;
ALTER TABLE STOCK20 ADD CONSTRAINT STOCK20CKC CHECK (S_W_ID
BETWEEN 16474 AND 17340);
SET INTEGRITY FOR STOCK20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK21 OFF;
ALTER TABLE STOCK21 DROP CONSTRAINT STOCK21CKC;
ALTER TABLE STOCK21 ADD CONSTRAINT STOCK21CKC CHECK (S_W_ID
BETWEEN 17341 AND 18207);
SET INTEGRITY FOR STOCK21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK22 OFF;
ALTER TABLE STOCK22 DROP CONSTRAINT STOCK22CKC;
ALTER TABLE STOCK22 ADD CONSTRAINT STOCK22CKC CHECK (S_W_ID
BETWEEN 18208 AND 19074);
SET INTEGRITY FOR STOCK22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK23 OFF;
ALTER TABLE STOCK23 DROP CONSTRAINT STOCK23CKC;
ALTER TABLE STOCK23 ADD CONSTRAINT STOCK23CKC CHECK (S_W_ID
BETWEEN 19075 AND 19941);
SET INTEGRITY FOR STOCK23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK24 OFF;
ALTER TABLE STOCK24 DROP CONSTRAINT STOCK24CKC;
ALTER TABLE STOCK24 ADD CONSTRAINT STOCK24CKC CHECK (S_W_ID
BETWEEN 19942 AND 20808);
SET INTEGRITY FOR STOCK24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK25 OFF;
ALTER TABLE STOCK25 DROP CONSTRAINT STOCK25CKC;
ALTER TABLE STOCK25 ADD CONSTRAINT STOCK25CKC CHECK (S_W_ID
BETWEEN 20809 AND 21675);
SET INTEGRITY FOR STOCK25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK26 OFF;
ALTER TABLE STOCK26 DROP CONSTRAINT STOCK26CKC;
ALTER TABLE STOCK26 ADD CONSTRAINT STOCK26CKC CHECK (S_W_ID
BETWEEN 21676 AND 22542);
SET INTEGRITY FOR STOCK26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK27 OFF;
ALTER TABLE STOCK27 DROP CONSTRAINT STOCK27CKC;
ALTER TABLE STOCK27 ADD CONSTRAINT STOCK27CKC CHECK (S_W_ID
BETWEEN 22543 AND 23409);
SET INTEGRITY FOR STOCK27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK28 OFF;
ALTER TABLE STOCK28 DROP CONSTRAINT STOCK28CKC;

```

```

ALTER TABLE STOCK28 ADD CONSTRAINT STOCK28CKC CHECK (S_W_ID
BETWEEN 23410 AND 24276);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK29 OFF;
ALTER TABLE STOCK29 DROP CONSTRAINT STOCK29CKC;
ALTER TABLE STOCK29 ADD CONSTRAINT STOCK29CKC CHECK (S_W_ID
BETWEEN 24277 AND 25143);
SET INTEGRITY FOR STOCK29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK30 OFF;
ALTER TABLE STOCK30 DROP CONSTRAINT STOCK30CKC;
ALTER TABLE STOCK30 ADD CONSTRAINT STOCK30CKC CHECK (S_W_ID
BETWEEN 25144 AND 26010);
SET INTEGRITY FOR STOCK30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK31 OFF;
ALTER TABLE STOCK31 DROP CONSTRAINT STOCK31CKC;
ALTER TABLE STOCK31 ADD CONSTRAINT STOCK31CKC CHECK (S_W_ID
BETWEEN 26011 AND 26877);
SET INTEGRITY FOR STOCK31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK32 OFF;
ALTER TABLE STOCK32 DROP CONSTRAINT STOCK32CKC;
ALTER TABLE STOCK32 ADD CONSTRAINT STOCK32CKC CHECK (S_W_ID
BETWEEN 26878 AND 27744);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK33 OFF;
ALTER TABLE STOCK33 DROP CONSTRAINT STOCK33CKC;
ALTER TABLE STOCK33 ADD CONSTRAINT STOCK33CKC CHECK (S_W_ID
BETWEEN 27745 AND 28611);
SET INTEGRITY FOR STOCK33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK34 OFF;
ALTER TABLE STOCK34 DROP CONSTRAINT STOCK34CKC;
ALTER TABLE STOCK34 ADD CONSTRAINT STOCK34CKC CHECK (S_W_ID
BETWEEN 28612 AND 29478);
SET INTEGRITY FOR STOCK34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK35 OFF;
ALTER TABLE STOCK35 DROP CONSTRAINT STOCK35CKC;
ALTER TABLE STOCK35 ADD CONSTRAINT STOCK35CKC CHECK (S_W_ID
BETWEEN 29479 AND 30345);
SET INTEGRITY FOR STOCK35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK36 OFF;
ALTER TABLE STOCK36 DROP CONSTRAINT STOCK36CKC;
ALTER TABLE STOCK36 ADD CONSTRAINT STOCK36CKC CHECK (S_W_ID
BETWEEN 30346 AND 31212);
SET INTEGRITY FOR STOCK36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK37 OFF;
ALTER TABLE STOCK37 DROP CONSTRAINT STOCK37CKC;
ALTER TABLE STOCK37 ADD CONSTRAINT STOCK37CKC CHECK (S_W_ID
BETWEEN 31213 AND 32079);
SET INTEGRITY FOR STOCK37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK38 OFF;
ALTER TABLE STOCK38 DROP CONSTRAINT STOCK38CKC;
ALTER TABLE STOCK38 ADD CONSTRAINT STOCK38CKC CHECK (S_W_ID
BETWEEN 32080 AND 32946);
SET INTEGRITY FOR STOCK38 ALL IMMEDIATE UNCHECKED;
connect reset;

```



```

ALTER TABLE STOCK70 ADD CONSTRAINT STOCK70CKC CHECK (S_W_ID
BETWEEN 59824 AND 60690);
SET INTEGRITY FOR STOCK70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK71 OFF;
ALTER TABLE STOCK71 DROP CONSTRAINT STOCK71CKC;
ALTER TABLE STOCK71 ADD CONSTRAINT STOCK71CKC CHECK (S_W_ID
BETWEEN 60691 AND 61557);
SET INTEGRITY FOR STOCK71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK72 OFF;
ALTER TABLE STOCK72 DROP CONSTRAINT STOCK72CKC;
ALTER TABLE STOCK72 ADD CONSTRAINT STOCK72CKC CHECK (S_W_ID
BETWEEN 61558 AND 62424);
SET INTEGRITY FOR STOCK72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK73 OFF;
ALTER TABLE STOCK73 DROP CONSTRAINT STOCK73CKC;
ALTER TABLE STOCK73 ADD CONSTRAINT STOCK73CKC CHECK (S_W_ID
BETWEEN 62425 AND 63291);
SET INTEGRITY FOR STOCK73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK74 OFF;
ALTER TABLE STOCK74 DROP CONSTRAINT STOCK74CKC;
ALTER TABLE STOCK74 ADD CONSTRAINT STOCK74CKC CHECK (S_W_ID
BETWEEN 63292 AND 64158);
SET INTEGRITY FOR STOCK74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK75 OFF;
ALTER TABLE STOCK75 DROP CONSTRAINT STOCK75CKC;
ALTER TABLE STOCK75 ADD CONSTRAINT STOCK75CKC CHECK (S_W_ID
BETWEEN 64159 AND 65025);
SET INTEGRITY FOR STOCK75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK76 OFF;
ALTER TABLE STOCK76 DROP CONSTRAINT STOCK76CKC;
ALTER TABLE STOCK76 ADD CONSTRAINT STOCK76CKC CHECK (S_W_ID
BETWEEN 65026 AND 65892);
SET INTEGRITY FOR STOCK76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK77 OFF;
ALTER TABLE STOCK77 DROP CONSTRAINT STOCK77CKC;
ALTER TABLE STOCK77 ADD CONSTRAINT STOCK77CKC CHECK (S_W_ID
BETWEEN 65893 AND 66759);
SET INTEGRITY FOR STOCK77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK78 OFF;
ALTER TABLE STOCK78 DROP CONSTRAINT STOCK78CKC;
ALTER TABLE STOCK78 ADD CONSTRAINT STOCK78CKC CHECK (S_W_ID
BETWEEN 66760 AND 67626);
SET INTEGRITY FOR STOCK78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK79 OFF;
ALTER TABLE STOCK79 DROP CONSTRAINT STOCK79CKC;
ALTER TABLE STOCK79 ADD CONSTRAINT STOCK79CKC CHECK (S_W_ID
BETWEEN 67627 AND 68493);
SET INTEGRITY FOR STOCK79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK80 OFF;
ALTER TABLE STOCK80 DROP CONSTRAINT STOCK80CKC;
ALTER TABLE STOCK80 ADD CONSTRAINT STOCK80CKC CHECK (S_W_ID
BETWEEN 68494 AND 69360);
SET INTEGRITY FOR STOCK80 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK81 OFF;
ALTER TABLE STOCK81 DROP CONSTRAINT STOCK81CKC;
ALTER TABLE STOCK81 ADD CONSTRAINT STOCK81CKC CHECK (S_W_ID
BETWEEN 69361 AND 70227);
SET INTEGRITY FOR STOCK81 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK82 OFF;
ALTER TABLE STOCK82 DROP CONSTRAINT STOCK82CKC;
ALTER TABLE STOCK82 ADD CONSTRAINT STOCK82CKC CHECK (S_W_ID
BETWEEN 70228 AND 71094);
SET INTEGRITY FOR STOCK82 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK83 OFF;
ALTER TABLE STOCK83 DROP CONSTRAINT STOCK83CKC;
ALTER TABLE STOCK83 ADD CONSTRAINT STOCK83CKC CHECK (S_W_ID
BETWEEN 71095 AND 71961);
SET INTEGRITY FOR STOCK83 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK84 OFF;
ALTER TABLE STOCK84 DROP CONSTRAINT STOCK84CKC;
ALTER TABLE STOCK84 ADD CONSTRAINT STOCK84CKC CHECK (S_W_ID
BETWEEN 71962 AND 72828);
SET INTEGRITY FOR STOCK84 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK85 OFF;
ALTER TABLE STOCK85 DROP CONSTRAINT STOCK85CKC;
ALTER TABLE STOCK85 ADD CONSTRAINT STOCK85CKC CHECK (S_W_ID
BETWEEN 72829 AND 73695);
SET INTEGRITY FOR STOCK85 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK86 OFF;
ALTER TABLE STOCK86 DROP CONSTRAINT STOCK86CKC;
ALTER TABLE STOCK86 ADD CONSTRAINT STOCK86CKC CHECK (S_W_ID
BETWEEN 73696 AND 74562);
SET INTEGRITY FOR STOCK86 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK87 OFF;
ALTER TABLE STOCK87 DROP CONSTRAINT STOCK87CKC;
ALTER TABLE STOCK87 ADD CONSTRAINT STOCK87CKC CHECK (S_W_ID
BETWEEN 74563 AND 75429);
SET INTEGRITY FOR STOCK87 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK88 OFF;
ALTER TABLE STOCK88 DROP CONSTRAINT STOCK88CKC;
ALTER TABLE STOCK88 ADD CONSTRAINT STOCK88CKC CHECK (S_W_ID
BETWEEN 75430 AND 76296);
SET INTEGRITY FOR STOCK88 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK89 OFF;
ALTER TABLE STOCK89 DROP CONSTRAINT STOCK89CKC;
ALTER TABLE STOCK89 ADD CONSTRAINT STOCK89CKC CHECK (S_W_ID
BETWEEN 76297 AND 77163);
SET INTEGRITY FOR STOCK89 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK90 OFF;
ALTER TABLE STOCK90 DROP CONSTRAINT STOCK90CKC;
ALTER TABLE STOCK90 ADD CONSTRAINT STOCK90CKC CHECK (S_W_ID
BETWEEN 77164 AND 78030);
SET INTEGRITY FOR STOCK90 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK91 OFF;
ALTER TABLE STOCK91 DROP CONSTRAINT STOCK91CKC;

```

```

ALTER TABLE STOCK91 ADD CONSTRAINT STOCK91CKC CHECK (S_W_ID
BETWEEN 78031 AND 78897);
SET INTEGRITY FOR STOCK91 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK92 OFF;
ALTER TABLE STOCK92 DROP CONSTRAINT STOCK92CKC;
ALTER TABLE STOCK92 ADD CONSTRAINT STOCK92CKC CHECK (S_W_ID
BETWEEN 78898 AND 79764);
SET INTEGRITY FOR STOCK92 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK93 OFF;
ALTER TABLE STOCK93 DROP CONSTRAINT STOCK93CKC;
ALTER TABLE STOCK93 ADD CONSTRAINT STOCK93CKC CHECK (S_W_ID
BETWEEN 79765 AND 80631);
SET INTEGRITY FOR STOCK93 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK94 OFF;
ALTER TABLE STOCK94 DROP CONSTRAINT STOCK94CKC;
ALTER TABLE STOCK94 ADD CONSTRAINT STOCK94CKC CHECK (S_W_ID
BETWEEN 80632 AND 81498);
SET INTEGRITY FOR STOCK94 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK95 OFF;
ALTER TABLE STOCK95 DROP CONSTRAINT STOCK95CKC;
ALTER TABLE STOCK95 ADD CONSTRAINT STOCK95CKC CHECK (S_W_ID
BETWEEN 81499 AND 82365);
SET INTEGRITY FOR STOCK95 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK96 OFF;
ALTER TABLE STOCK96 DROP CONSTRAINT STOCK96CKC;
ALTER TABLE STOCK96 ADD CONSTRAINT STOCK96CKC CHECK (S_W_ID
BETWEEN 82366 AND 83232);
SET INTEGRITY FOR STOCK96 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK97 OFF;
ALTER TABLE STOCK97 DROP CONSTRAINT STOCK97CKC;
ALTER TABLE STOCK97 ADD CONSTRAINT STOCK97CKC CHECK (S_W_ID
BETWEEN 83233 AND 84099);
SET INTEGRITY FOR STOCK97 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK98 OFF;
ALTER TABLE STOCK98 DROP CONSTRAINT STOCK98CKC;
ALTER TABLE STOCK98 ADD CONSTRAINT STOCK98CKC CHECK (S_W_ID
BETWEEN 84100 AND 84966);
SET INTEGRITY FOR STOCK98 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK99 OFF;
ALTER TABLE STOCK99 DROP CONSTRAINT STOCK99CKC;
ALTER TABLE STOCK99 ADD CONSTRAINT STOCK99CKC CHECK (S_W_ID
BETWEEN 84967 AND 85833);
SET INTEGRITY FOR STOCK99 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK100 OFF;
ALTER TABLE STOCK100 DROP CONSTRAINT STOCK100CKC;
ALTER TABLE STOCK100 ADD CONSTRAINT STOCK100CKC CHECK (S_W_ID
BETWEEN 85834 AND 86700);
SET INTEGRITY FOR STOCK100 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK101 OFF;
ALTER TABLE STOCK101 DROP CONSTRAINT STOCK101CKC;
ALTER TABLE STOCK101 ADD CONSTRAINT STOCK101CKC CHECK (S_W_ID
BETWEEN 86701 AND 87567);
SET INTEGRITY FOR STOCK101 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK102 OFF;
ALTER TABLE STOCK102 DROP CONSTRAINT STOCK102CKC;
ALTER TABLE STOCK102 ADD CONSTRAINT STOCK102CKC CHECK (S_W_ID
BETWEEN 87568 AND 88434);
SET INTEGRITY FOR STOCK102 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK103 OFF;
ALTER TABLE STOCK103 DROP CONSTRAINT STOCK103CKC;
ALTER TABLE STOCK103 ADD CONSTRAINT STOCK103CKC CHECK (S_W_ID
BETWEEN 88435 AND 89301);
SET INTEGRITY FOR STOCK103 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK104 OFF;
ALTER TABLE STOCK104 DROP CONSTRAINT STOCK104CKC;
ALTER TABLE STOCK104 ADD CONSTRAINT STOCK104CKC CHECK (S_W_ID
BETWEEN 89302 AND 90168);
SET INTEGRITY FOR STOCK104 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK105 OFF;
ALTER TABLE STOCK105 DROP CONSTRAINT STOCK105CKC;
ALTER TABLE STOCK105 ADD CONSTRAINT STOCK105CKC CHECK (S_W_ID
BETWEEN 90169 AND 91035);
SET INTEGRITY FOR STOCK105 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK106 OFF;
ALTER TABLE STOCK106 DROP CONSTRAINT STOCK106CKC;
ALTER TABLE STOCK106 ADD CONSTRAINT STOCK106CKC CHECK (S_W_ID
BETWEEN 91036 AND 91902);
SET INTEGRITY FOR STOCK106 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK107 OFF;
ALTER TABLE STOCK107 DROP CONSTRAINT STOCK107CKC;
ALTER TABLE STOCK107 ADD CONSTRAINT STOCK107CKC CHECK (S_W_ID
BETWEEN 91903 AND 92769);
SET INTEGRITY FOR STOCK107 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK108 OFF;
ALTER TABLE STOCK108 DROP CONSTRAINT STOCK108CKC;
ALTER TABLE STOCK108 ADD CONSTRAINT STOCK108CKC CHECK (S_W_ID
BETWEEN 92770 AND 93636);
SET INTEGRITY FOR STOCK108 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK109 OFF;
ALTER TABLE STOCK109 DROP CONSTRAINT STOCK109CKC;
ALTER TABLE STOCK109 ADD CONSTRAINT STOCK109CKC CHECK (S_W_ID
BETWEEN 93637 AND 94503);
SET INTEGRITY FOR STOCK109 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK110 OFF;
ALTER TABLE STOCK110 DROP CONSTRAINT STOCK110CKC;
ALTER TABLE STOCK110 ADD CONSTRAINT STOCK110CKC CHECK (S_W_ID BETWEEN 94504
AND 95370);
SET INTEGRITY FOR STOCK110 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK111 OFF;
ALTER TABLE STOCK111 DROP CONSTRAINT STOCK111CKC;
ALTER TABLE STOCK111 ADD CONSTRAINT STOCK111CKC CHECK (S_W_ID BETWEEN 95371
AND 96237);
SET INTEGRITY FOR STOCK111 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK112 OFF;
ALTER TABLE STOCK112 DROP CONSTRAINT STOCK112CKC;

```

```

ALTER TABLE STOCK112 ADD CONSTRAINT STOCK112CKC CHECK (S_W_ID BETWEEN 96238
AND 97104);
SET INTEGRITY FOR STOCK112 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK113 OFF;
ALTER TABLE STOCK113 DROP CONSTRAINT STOCK113CKC;
ALTER TABLE STOCK113 ADD CONSTRAINT STOCK113CKC CHECK (S_W_ID BETWEEN 97105
AND 97971);
SET INTEGRITY FOR STOCK113 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK114 OFF;
ALTER TABLE STOCK114 DROP CONSTRAINT STOCK114CKC;
ALTER TABLE STOCK114 ADD CONSTRAINT STOCK114CKC CHECK (S_W_ID BETWEEN 97972
AND 98838);
SET INTEGRITY FOR STOCK114 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK115 OFF;
ALTER TABLE STOCK115 DROP CONSTRAINT STOCK115CKC;
ALTER TABLE STOCK115 ADD CONSTRAINT STOCK115CKC CHECK (S_W_ID BETWEEN 98839
AND 99705);
SET INTEGRITY FOR STOCK115 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK116 OFF;
ALTER TABLE STOCK116 DROP CONSTRAINT STOCK116CKC;
ALTER TABLE STOCK116 ADD CONSTRAINT STOCK116CKC CHECK (S_W_ID BETWEEN 99706
AND 100572);
SET INTEGRITY FOR STOCK116 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK117 OFF;
ALTER TABLE STOCK117 DROP CONSTRAINT STOCK117CKC;
ALTER TABLE STOCK117 ADD CONSTRAINT STOCK117CKC CHECK (S_W_ID BETWEEN 100573
AND 101439);
SET INTEGRITY FOR STOCK117 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK118 OFF;
ALTER TABLE STOCK118 DROP CONSTRAINT STOCK118CKC;
ALTER TABLE STOCK118 ADD CONSTRAINT STOCK118CKC CHECK (S_W_ID BETWEEN 101440
AND 102306);
SET INTEGRITY FOR STOCK118 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK119 OFF;
ALTER TABLE STOCK119 DROP CONSTRAINT STOCK119CKC;
ALTER TABLE STOCK119 ADD CONSTRAINT STOCK119CKC CHECK (S_W_ID BETWEEN 102307
AND 103173);
SET INTEGRITY FOR STOCK119 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK120 OFF;
ALTER TABLE STOCK120 DROP CONSTRAINT STOCK120CKC;
ALTER TABLE STOCK120 ADD CONSTRAINT STOCK120CKC CHECK (S_W_ID >= 103174);
SET INTEGRITY FOR STOCK120 ALL IMMEDIATE UNCHECKED;
connect reset;

```

### **DDL/CRCONST WAREHOUSE.ddl**

```

connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE1 OFF;
ALTER TABLE WAREHOUSE1 DROP CONSTRAINT WAREHOUSE1CKC;
ALTER TABLE WAREHOUSE1 ADD CONSTRAINT WAREHOUSE1CKC CHECK (W_ID BETWEEN 1
AND 867);
SET INTEGRITY FOR WAREHOUSE1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE2 OFF;

```

```

ALTER TABLE WAREHOUSE2 DROP CONSTRAINT WAREHOUSE2CKC;
ALTER TABLE WAREHOUSE2 ADD CONSTRAINT WAREHOUSE2CKC CHECK (W_ID BETWEEN 868
AND 1734);
SET INTEGRITY FOR WAREHOUSE2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE3 OFF;
ALTER TABLE WAREHOUSE3 DROP CONSTRAINT WAREHOUSE3CKC;
ALTER TABLE WAREHOUSE3 ADD CONSTRAINT WAREHOUSE3CKC CHECK (W_ID BETWEEN
1735 AND 2601);
SET INTEGRITY FOR WAREHOUSE3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE4 OFF;
ALTER TABLE WAREHOUSE4 DROP CONSTRAINT WAREHOUSE4CKC;
ALTER TABLE WAREHOUSE4 ADD CONSTRAINT WAREHOUSE4CKC CHECK (W_ID BETWEEN
2602 AND 3468);
SET INTEGRITY FOR WAREHOUSE4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE5 OFF;
ALTER TABLE WAREHOUSE5 DROP CONSTRAINT WAREHOUSE5CKC;
ALTER TABLE WAREHOUSE5 ADD CONSTRAINT WAREHOUSE5CKC CHECK (W_ID BETWEEN
3469 AND 4335);
SET INTEGRITY FOR WAREHOUSE5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE6 OFF;
ALTER TABLE WAREHOUSE6 DROP CONSTRAINT WAREHOUSE6CKC;
ALTER TABLE WAREHOUSE6 ADD CONSTRAINT WAREHOUSE6CKC CHECK (W_ID BETWEEN
4336 AND 5202);
SET INTEGRITY FOR WAREHOUSE6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE7 OFF;
ALTER TABLE WAREHOUSE7 DROP CONSTRAINT WAREHOUSE7CKC;
ALTER TABLE WAREHOUSE7 ADD CONSTRAINT WAREHOUSE7CKC CHECK (W_ID BETWEEN
5203 AND 6069);
SET INTEGRITY FOR WAREHOUSE7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE8 OFF;
ALTER TABLE WAREHOUSE8 DROP CONSTRAINT WAREHOUSE8CKC;
ALTER TABLE WAREHOUSE8 ADD CONSTRAINT WAREHOUSE8CKC CHECK (W_ID BETWEEN
6070 AND 6936);
SET INTEGRITY FOR WAREHOUSE8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE9 OFF;
ALTER TABLE WAREHOUSE9 DROP CONSTRAINT WAREHOUSE9CKC;
ALTER TABLE WAREHOUSE9 ADD CONSTRAINT WAREHOUSE9CKC CHECK (W_ID BETWEEN
6937 AND 7803);
SET INTEGRITY FOR WAREHOUSE9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE10 OFF;
ALTER TABLE WAREHOUSE10 DROP CONSTRAINT WAREHOUSE10CKC;
ALTER TABLE WAREHOUSE10 ADD CONSTRAINT WAREHOUSE10CKC CHECK (W_ID BETWEEN
7804 AND 8670);
SET INTEGRITY FOR WAREHOUSE10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE11 OFF;
ALTER TABLE WAREHOUSE11 DROP CONSTRAINT WAREHOUSE11CKC;
ALTER TABLE WAREHOUSE11 ADD CONSTRAINT WAREHOUSE11CKC CHECK (W_ID BETWEEN
8671 AND 9537);
SET INTEGRITY FOR WAREHOUSE11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE12 OFF;
ALTER TABLE WAREHOUSE12 DROP CONSTRAINT WAREHOUSE12CKC;
ALTER TABLE WAREHOUSE12 ADD CONSTRAINT WAREHOUSE12CKC CHECK (W_ID BETWEEN
9538 AND 10404);

```









```

ALTER TABLE WAREHOUSE106 DROP CONSTRAINT WAREHOUSE106CKC;
ALTER TABLE WAREHOUSE106 ADD CONSTRAINT WAREHOUSE106CKC CHECK (W_ID BETWEEN
91036 AND 91902);
SET INTEGRITY FOR WAREHOUSE106 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE107 OFF;
ALTER TABLE WAREHOUSE107 DROP CONSTRAINT WAREHOUSE107CKC;
ALTER TABLE WAREHOUSE107 ADD CONSTRAINT WAREHOUSE107CKC CHECK (W_ID BETWEEN
91903 AND 92769);
SET INTEGRITY FOR WAREHOUSE107 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE108 OFF;
ALTER TABLE WAREHOUSE108 DROP CONSTRAINT WAREHOUSE108CKC;
ALTER TABLE WAREHOUSE108 ADD CONSTRAINT WAREHOUSE108CKC CHECK (W_ID BETWEEN
92770 AND 93636);
SET INTEGRITY FOR WAREHOUSE108 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE109 OFF;
ALTER TABLE WAREHOUSE109 DROP CONSTRAINT WAREHOUSE109CKC;
ALTER TABLE WAREHOUSE109 ADD CONSTRAINT WAREHOUSE109CKC CHECK (W_ID BETWEEN
93637 AND 94503);
SET INTEGRITY FOR WAREHOUSE109 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE110 OFF;
ALTER TABLE WAREHOUSE110 DROP CONSTRAINT WAREHOUSE110CKC;
ALTER TABLE WAREHOUSE110 ADD CONSTRAINT WAREHOUSE110CKC CHECK (W_ID BETWEEN
94504 AND 95370);
SET INTEGRITY FOR WAREHOUSE110 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE111 OFF;
ALTER TABLE WAREHOUSE111 DROP CONSTRAINT WAREHOUSE111CKC;
ALTER TABLE WAREHOUSE111 ADD CONSTRAINT WAREHOUSE111CKC CHECK (W_ID BETWEEN
95371 AND 96237);
SET INTEGRITY FOR WAREHOUSE111 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE112 OFF;
ALTER TABLE WAREHOUSE112 DROP CONSTRAINT WAREHOUSE112CKC;
ALTER TABLE WAREHOUSE112 ADD CONSTRAINT WAREHOUSE112CKC CHECK (W_ID BETWEEN
96238 AND 97104);
SET INTEGRITY FOR WAREHOUSE112 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE113 OFF;
ALTER TABLE WAREHOUSE113 DROP CONSTRAINT WAREHOUSE113CKC;
ALTER TABLE WAREHOUSE113 ADD CONSTRAINT WAREHOUSE113CKC CHECK (W_ID BETWEEN
97105 AND 97971);
SET INTEGRITY FOR WAREHOUSE113 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE114 OFF;
ALTER TABLE WAREHOUSE114 DROP CONSTRAINT WAREHOUSE114CKC;
ALTER TABLE WAREHOUSE114 ADD CONSTRAINT WAREHOUSE114CKC CHECK (W_ID BETWEEN
97972 AND 98838);
SET INTEGRITY FOR WAREHOUSE114 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE115 OFF;
ALTER TABLE WAREHOUSE115 DROP CONSTRAINT WAREHOUSE115CKC;
ALTER TABLE WAREHOUSE115 ADD CONSTRAINT WAREHOUSE115CKC CHECK (W_ID BETWEEN
98839 AND 99705);
SET INTEGRITY FOR WAREHOUSE115 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE116 OFF;
ALTER TABLE WAREHOUSE116 DROP CONSTRAINT WAREHOUSE116CKC;
ALTER TABLE WAREHOUSE116 ADD CONSTRAINT WAREHOUSE116CKC CHECK (W_ID BETWEEN
99706 AND 100572);

```

```

SET INTEGRITY FOR WAREHOUSE116 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE117 OFF;
ALTER TABLE WAREHOUSE117 DROP CONSTRAINT WAREHOUSE117CKC;
ALTER TABLE WAREHOUSE117 ADD CONSTRAINT WAREHOUSE117CKC CHECK (W_ID BETWEEN
100573 AND 101439);
SET INTEGRITY FOR WAREHOUSE117 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE118 OFF;
ALTER TABLE WAREHOUSE118 DROP CONSTRAINT WAREHOUSE118CKC;
ALTER TABLE WAREHOUSE118 ADD CONSTRAINT WAREHOUSE118CKC CHECK (W_ID BETWEEN
101440 AND 102306);
SET INTEGRITY FOR WAREHOUSE118 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE119 OFF;
ALTER TABLE WAREHOUSE119 DROP CONSTRAINT WAREHOUSE119CKC;
ALTER TABLE WAREHOUSE119 ADD CONSTRAINT WAREHOUSE119CKC CHECK (W_ID BETWEEN
102307 AND 103173);
SET INTEGRITY FOR WAREHOUSE119 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE120 OFF;
ALTER TABLE WAREHOUSE120 DROP CONSTRAINT WAREHOUSE120CKC;
ALTER TABLE WAREHOUSE120 ADD CONSTRAINT WAREHOUSE120CKC CHECK (W_ID >=
103174);
SET INTEGRITY FOR WAREHOUSE120 ALL IMMEDIATE UNCHECKED;
connect reset;

```

### **DDL/CRIDX CUST IDXB.ddl**

```

connect to TPCC in share mode;
DROP INDEX CUST_IDXB1;
CREATE INDEX CUST_IDXB1
ON CUSTOMER1(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB2;
CREATE INDEX CUST_IDXB2
ON CUSTOMER2(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB3;
CREATE INDEX CUST_IDXB3
ON CUSTOMER3(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB4;
CREATE INDEX CUST_IDXB4
ON CUSTOMER4(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB5;
CREATE INDEX CUST_IDXB5
ON CUSTOMER5(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB6;
CREATE INDEX CUST_IDXB6
ON CUSTOMER6(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB7;
CREATE INDEX CUST_IDXB7
ON CUSTOMER7(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB8;
CREATE INDEX CUST_IDXB8

```

```

ON CUSTOMER8(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB9;
CREATE INDEX CUST_IDXB9
ON CUSTOMER9(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB10;
CREATE INDEX CUST_IDXB10
ON CUSTOMER10(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB11;
CREATE INDEX CUST_IDXB11
ON CUSTOMER11(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB12;
CREATE INDEX CUST_IDXB12
ON CUSTOMER12(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB13;
CREATE INDEX CUST_IDXB13
ON CUSTOMER13(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB14;
CREATE INDEX CUST_IDXB14
ON CUSTOMER14(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB15;
CREATE INDEX CUST_IDXB15
ON CUSTOMER15(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB16;
CREATE INDEX CUST_IDXB16
ON CUSTOMER16(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB17;
CREATE INDEX CUST_IDXB17
ON CUSTOMER17(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB18;
CREATE INDEX CUST_IDXB18
ON CUSTOMER18(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB19;
CREATE INDEX CUST_IDXB19
ON CUSTOMER19(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB20;
CREATE INDEX CUST_IDXB20
ON CUSTOMER20(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB21;
CREATE INDEX CUST_IDXB21
ON CUSTOMER21(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB22;
CREATE INDEX CUST_IDXB22
ON CUSTOMER22(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;

```













## DDL/CRTB\_CUSTOMER.ddl

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER1;
CREATE TABLE CUSTOMER1
```

```
(
  C_ID          INTEGER    NOT NULL,
  C_STATE       CHAR(2)    NOT NULL,
  C_ZIP         CHAR(9)    NOT NULL,
  C_PHONE       CHAR(16)   NOT NULL,
  C_SINCE       TIMESTAMP  NOT NULL,
  C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
  C_MIDDLE      CHAR(2)    NOT NULL,
  C_CREDIT      CHAR(2)    NOT NULL,
  C_DISCOUNT   REAL      NOT NULL,
  C_DATA        VARCHAR(500) NOT NULL,
  C_LAST        VARCHAR(16) NOT NULL,
  C_FIRST       VARCHAR(16) NOT NULL,
  C_STREET_1    VARCHAR(20) NOT NULL,
  C_STREET_2    VARCHAR(20) NOT NULL,
  C_CITY        VARCHAR(20) NOT NULL,
  C_D_ID        SMALLINT   NOT NULL,
  C_W_ID        INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE     DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_001
INDEX IN is_customer_001
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 1 ENDING AT 867,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER2;
CREATE TABLE CUSTOMER2
```

```
(
  C_ID          INTEGER    NOT NULL,
  C_STATE       CHAR(2)    NOT NULL,
  C_ZIP         CHAR(9)    NOT NULL,
  C_PHONE       CHAR(16)   NOT NULL,
  C_SINCE       TIMESTAMP  NOT NULL,
  C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
  C_MIDDLE      CHAR(2)    NOT NULL,
  C_CREDIT      CHAR(2)    NOT NULL,
  C_DISCOUNT   REAL      NOT NULL,
  C_DATA        VARCHAR(500) NOT NULL,
  C_LAST        VARCHAR(16) NOT NULL,
  C_FIRST       VARCHAR(16) NOT NULL,
  C_STREET_1    VARCHAR(20) NOT NULL,
  C_STREET_2    VARCHAR(20) NOT NULL,
  C_CITY        VARCHAR(20) NOT NULL,
  C_D_ID        SMALLINT   NOT NULL,
  C_W_ID        INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE     DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_002
INDEX IN is_customer_002
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 868 ENDING AT 1734,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
```

```
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER3;
CREATE TABLE CUSTOMER3
```

```
(
  C_ID          INTEGER    NOT NULL,
  C_STATE       CHAR(2)    NOT NULL,
  C_ZIP         CHAR(9)    NOT NULL,
  C_PHONE       CHAR(16)   NOT NULL,
  C_SINCE       TIMESTAMP  NOT NULL,
  C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
  C_MIDDLE      CHAR(2)    NOT NULL,
  C_CREDIT      CHAR(2)    NOT NULL,
  C_DISCOUNT   REAL      NOT NULL,
  C_DATA        VARCHAR(500) NOT NULL,
  C_LAST        VARCHAR(16) NOT NULL,
  C_FIRST       VARCHAR(16) NOT NULL,
  C_STREET_1    VARCHAR(20) NOT NULL,
  C_STREET_2    VARCHAR(20) NOT NULL,
  C_CITY        VARCHAR(20) NOT NULL,
  C_D_ID        SMALLINT   NOT NULL,
  C_W_ID        INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE     DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_003
INDEX IN is_customer_003
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 1735 ENDING AT 2601,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER4;
CREATE TABLE CUSTOMER4
```

```
(
  C_ID          INTEGER    NOT NULL,
  C_STATE       CHAR(2)    NOT NULL,
  C_ZIP         CHAR(9)    NOT NULL,
  C_PHONE       CHAR(16)   NOT NULL,
  C_SINCE       TIMESTAMP  NOT NULL,
  C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
  C_MIDDLE      CHAR(2)    NOT NULL,
  C_CREDIT      CHAR(2)    NOT NULL,
  C_DISCOUNT   REAL      NOT NULL,
  C_DATA        VARCHAR(500) NOT NULL,
  C_LAST        VARCHAR(16) NOT NULL,
  C_FIRST       VARCHAR(16) NOT NULL,
  C_STREET_1    VARCHAR(20) NOT NULL,
  C_STREET_2    VARCHAR(20) NOT NULL,
  C_CITY        VARCHAR(20) NOT NULL,
  C_D_ID        SMALLINT   NOT NULL,
  C_W_ID        INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE     DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_004
INDEX IN is_customer_004
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 2602 ENDING AT 3468,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER5;
CREATE TABLE CUSTOMER5
```

```
(
  C_ID          INTEGER    NOT NULL,
  C_STATE       CHAR(2)    NOT NULL,
  C_ZIP         CHAR(9)    NOT NULL,
  C_PHONE       CHAR(16)   NOT NULL,
  C_SINCE       TIMESTAMP  NOT NULL,
  C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
  C_MIDDLE      CHAR(2)    NOT NULL,
  C_CREDIT      CHAR(2)    NOT NULL,
  C_DISCOUNT   REAL      NOT NULL,
  C_DATA        VARCHAR(500) NOT NULL,
  C_LAST        VARCHAR(16) NOT NULL,
  C_FIRST       VARCHAR(16) NOT NULL,
  C_STREET_1    VARCHAR(20) NOT NULL,
  C_STREET_2    VARCHAR(20) NOT NULL,
  C_CITY        VARCHAR(20) NOT NULL,
  C_D_ID        SMALLINT   NOT NULL,
  C_W_ID        INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE     DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_005
INDEX IN is_customer_005
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 3469 ENDING AT 4335,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER6;
CREATE TABLE CUSTOMER6
```

```
(
  C_ID          INTEGER    NOT NULL,
  C_STATE       CHAR(2)    NOT NULL,
  C_ZIP         CHAR(9)    NOT NULL,
  C_PHONE       CHAR(16)   NOT NULL,
  C_SINCE       TIMESTAMP  NOT NULL,
  C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
  C_MIDDLE      CHAR(2)    NOT NULL,
  C_CREDIT      CHAR(2)    NOT NULL,
  C_DISCOUNT   REAL      NOT NULL,
  C_DATA        VARCHAR(500) NOT NULL,
  C_LAST        VARCHAR(16) NOT NULL,
  C_FIRST       VARCHAR(16) NOT NULL,
  C_STREET_1    VARCHAR(20) NOT NULL,
  C_STREET_2    VARCHAR(20) NOT NULL,
  C_CITY        VARCHAR(20) NOT NULL,
  C_D_ID        SMALLINT   NOT NULL,
  C_W_ID        INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE     DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_006
INDEX IN is_customer_006
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 4336 ENDING AT 5202,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
```

```
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER7;
CREATE TABLE CUSTOMER7
```

```
(
  C_ID          INTEGER    NOT NULL,
  C_STATE       CHAR(2)    NOT NULL,
```

```

C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_007
INDEX IN is_customer_007
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 5203 ENDING AT 6069,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER8;
CREATE TABLE CUSTOMER8
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_008
INDEX IN is_customer_008
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 6070 ENDING AT 6936,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER9;
CREATE TABLE CUSTOMER9
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,

```

```

C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_009
INDEX IN is_customer_009
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 6937 ENDING AT 7803,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER10;
CREATE TABLE CUSTOMER10
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_010
INDEX IN is_customer_010
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 7804 ENDING AT 8670,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER11;
CREATE TABLE CUSTOMER11
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,

```

```

C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_011
INDEX IN is_customer_011
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 8671 ENDING AT 9537,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER12;
CREATE TABLE CUSTOMER12
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_012
INDEX IN is_customer_012
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 9538 ENDING AT 10404,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER13;
CREATE TABLE CUSTOMER13
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,

```

```

C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_013
INDEX IN is_customer_013
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 10405 ENDING AT 11271,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER14;
CREATE TABLE CUSTOMER14
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_014
INDEX IN is_customer_014
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 11272 ENDING AT 12138,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER15;
CREATE TABLE CUSTOMER15
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,

```

```

C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_015
INDEX IN is_customer_015
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 12139 ENDING AT 13005,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER16;
CREATE TABLE CUSTOMER16
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_016
INDEX IN is_customer_016
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 13006 ENDING AT 13872,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER17;
CREATE TABLE CUSTOMER17
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,

```

```

C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_017
INDEX IN is_customer_017
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 13873 ENDING AT 14739,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER18;
CREATE TABLE CUSTOMER18
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_018
INDEX IN is_customer_018
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 14740 ENDING AT 15606,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER19;
CREATE TABLE CUSTOMER19
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,

```

```

C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_019
INDEX IN is_customer_019
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 15607 ENDING AT 16473,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER20;
CREATE TABLE CUSTOMER20
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_020
INDEX IN is_customer_020
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 16474 ENDING AT 17340,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER21;
CREATE TABLE CUSTOMER21
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_021

```

```

INDEX IN is_customer_021
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 17341 ENDING AT 18207,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER22;
CREATE TABLE CUSTOMER22
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_022
INDEX IN is_customer_022
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 18208 ENDING AT 19074,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER23;
CREATE TABLE CUSTOMER23
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_023
INDEX IN is_customer_023
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 19075 ENDING AT 19941,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER24;
CREATE TABLE CUSTOMER24
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_024
INDEX IN is_customer_024
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 19942 ENDING AT 20808,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER25;
CREATE TABLE CUSTOMER25
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_025
INDEX IN is_customer_025
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 20809 ENDING AT 21675,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER26;
CREATE TABLE CUSTOMER26
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_026
INDEX IN is_customer_026
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 21676 ENDING AT 22542,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER27;
CREATE TABLE CUSTOMER27
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_027
INDEX IN is_customer_027
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 22543 ENDING AT 23409,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE CUSTOMER28;
CREATE TABLE CUSTOMER28
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_028
INDEX IN is_customer_028
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 23410 ENDING AT 24276,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER29;
CREATE TABLE CUSTOMER29
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_029
INDEX IN is_customer_029
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 24277 ENDING AT 25143,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER30;
CREATE TABLE CUSTOMER30
(

```

```

  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_030
INDEX IN is_customer_030
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 25144 ENDING AT 26010,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER31;
CREATE TABLE CUSTOMER31
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER    NOT NULL
)
IN ts_customer_031
INDEX IN is_customer_031
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 26011 ENDING AT 26877,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER32;
CREATE TABLE CUSTOMER32
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,

```

```

C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_032
INDEX IN is_customer_032
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 26878 ENDING AT 27744,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER33;
CREATE TABLE CUSTOMER33
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_033
INDEX IN is_customer_033
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 27745 ENDING AT 28611,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER34;
CREATE TABLE CUSTOMER34
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,

```

```

C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_034
INDEX IN is_customer_034
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 28612 ENDING AT 29478,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER35;
CREATE TABLE CUSTOMER35
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_035
INDEX IN is_customer_035
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 29479 ENDING AT 30345,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER36;
CREATE TABLE CUSTOMER36
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,

```

```

C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_036
INDEX IN is_customer_036
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 30346 ENDING AT 31212,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER37;
CREATE TABLE CUSTOMER37
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_037
INDEX IN is_customer_037
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 31213 ENDING AT 32079,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER38;
CREATE TABLE CUSTOMER38
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,

```

```

C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_038
INDEX IN is_customer_038
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 32080 ENDING AT 32946,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER39;
CREATE TABLE CUSTOMER39
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_039
INDEX IN is_customer_039
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 32947 ENDING AT 33813,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER40;
CREATE TABLE CUSTOMER40
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,

```

```

C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_040
INDEX IN is_customer_040
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 33814 ENDING AT 34680,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER41;
CREATE TABLE CUSTOMER41
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_041
INDEX IN is_customer_041
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 34681 ENDING AT 35547,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER42;
CREATE TABLE CUSTOMER42
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,

```

```

C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_042
INDEX IN is_customer_042
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 35548 ENDING AT 36414,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER43;
CREATE TABLE CUSTOMER43
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_043
INDEX IN is_customer_043
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 36415 ENDING AT 37281,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER44;
CREATE TABLE CUSTOMER44
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL

```

```

)
IN ts_customer_044
INDEX IN is_customer_044
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 37282 ENDING AT 38148,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER45;
CREATE TABLE CUSTOMER45
(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_045
INDEX IN is_customer_045
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 38149 ENDING AT 39015,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER46;
CREATE TABLE CUSTOMER46
(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_046
INDEX IN is_customer_046

```

```

ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 39016 ENDING AT 39882,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER47;
CREATE TABLE CUSTOMER47
(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_047
INDEX IN is_customer_047
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 39883 ENDING AT 40749,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER48;
CREATE TABLE CUSTOMER48
(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_048
INDEX IN is_customer_048
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 40750 ENDING AT 41616,

```

```

  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER49;
CREATE TABLE CUSTOMER49
(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_049
INDEX IN is_customer_049
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 41617 ENDING AT 42483,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER50;
CREATE TABLE CUSTOMER50
(
  C_ID INTEGER NOT NULL,
  C_STATE CHAR(2) NOT NULL,
  C_ZIP CHAR(9) NOT NULL,
  C_PHONE CHAR(16) NOT NULL,
  C_SINCE TIMESTAMP NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE CHAR(2) NOT NULL,
  C_CREDIT CHAR(2) NOT NULL,
  C_DISCOUNT REAL NOT NULL,
  C_DATA VARCHAR(500) NOT NULL,
  C_LAST VARCHAR(16) NOT NULL,
  C_FIRST VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY VARCHAR(20) NOT NULL,
  C_D_ID SMALLINT NOT NULL,
  C_W_ID INTEGER NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_050
INDEX IN is_customer_050
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 42484 ENDING AT 43350,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```



```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER51;
CREATE TABLE CUSTOMER51
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500)  NOT NULL,
  C_LAST       VARCHAR(16)   NOT NULL,
  C_FIRST      VARCHAR(16)   NOT NULL,
  C_STREET_1   VARCHAR(20)   NOT NULL,
  C_STREET_2   VARCHAR(20)   NOT NULL,
  C_CITY       VARCHAR(20)   NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER       NOT NULL,
  C_DELIVERY_CNT INTEGER     NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_051
INDEX IN is_customer_051
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 43351 ENDING AT 44217,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER52;
CREATE TABLE CUSTOMER52

```

```

(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500)  NOT NULL,
  C_LAST       VARCHAR(16)   NOT NULL,
  C_FIRST      VARCHAR(16)   NOT NULL,
  C_STREET_1   VARCHAR(20)   NOT NULL,
  C_STREET_2   VARCHAR(20)   NOT NULL,
  C_CITY       VARCHAR(20)   NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER       NOT NULL,
  C_DELIVERY_CNT INTEGER     NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_052
INDEX IN is_customer_052
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 44218 ENDING AT 45084,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER53;

```

```

CREATE TABLE CUSTOMER53
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500)  NOT NULL,
  C_LAST       VARCHAR(16)   NOT NULL,
  C_FIRST      VARCHAR(16)   NOT NULL,
  C_STREET_1   VARCHAR(20)   NOT NULL,
  C_STREET_2   VARCHAR(20)   NOT NULL,
  C_CITY       VARCHAR(20)   NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER       NOT NULL,
  C_DELIVERY_CNT INTEGER     NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_053
INDEX IN is_customer_053
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 45085 ENDING AT 45951,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER54;
CREATE TABLE CUSTOMER54

```

```

(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500)  NOT NULL,
  C_LAST       VARCHAR(16)   NOT NULL,
  C_FIRST      VARCHAR(16)   NOT NULL,
  C_STREET_1   VARCHAR(20)   NOT NULL,
  C_STREET_2   VARCHAR(20)   NOT NULL,
  C_CITY       VARCHAR(20)   NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER       NOT NULL,
  C_DELIVERY_CNT INTEGER     NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_054
INDEX IN is_customer_054
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 45952 ENDING AT 46818,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER55;
CREATE TABLE CUSTOMER55

```

```

(
  C_ID          INTEGER      NOT NULL,

```

```

  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500)  NOT NULL,
  C_LAST       VARCHAR(16)   NOT NULL,
  C_FIRST      VARCHAR(16)   NOT NULL,
  C_STREET_1   VARCHAR(20)   NOT NULL,
  C_STREET_2   VARCHAR(20)   NOT NULL,
  C_CITY       VARCHAR(20)   NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER       NOT NULL,
  C_DELIVERY_CNT INTEGER     NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_055
INDEX IN is_customer_055
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 46819 ENDING AT 47685,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER56;
CREATE TABLE CUSTOMER56

```

```

(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500)  NOT NULL,
  C_LAST       VARCHAR(16)   NOT NULL,
  C_FIRST      VARCHAR(16)   NOT NULL,
  C_STREET_1   VARCHAR(20)   NOT NULL,
  C_STREET_2   VARCHAR(20)   NOT NULL,
  C_CITY       VARCHAR(20)   NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER       NOT NULL,
  C_DELIVERY_CNT INTEGER     NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_056
INDEX IN is_customer_056
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 47686 ENDING AT 48552,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER57;
CREATE TABLE CUSTOMER57

```

```

(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,

```

```

C_SINCE    TIMESTAMP    NOT NULL,
C_CREDIT_LIM  DECIMAL(12,2) NOT NULL,
C_MIDDLE    CHAR(2)    NOT NULL,
C_CREDIT    CHAR(2)    NOT NULL,
C_DISCOUNT REAL    NOT NULL,
C_DATA      VARCHAR(500) NOT NULL,
C_LAST      VARCHAR(16) NOT NULL,
C_FIRST     VARCHAR(16) NOT NULL,
C_STREET_1  VARCHAR(20) NOT NULL,
C_STREET_2  VARCHAR(20) NOT NULL,
C_CITY      VARCHAR(20) NOT NULL,
C_D_ID      SMALLINT   NOT NULL,
C_W_ID      INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE   DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_057
INDEX IN is_customer_057
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 48553 ENDING AT 49419,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER58;
CREATE TABLE CUSTOMER58
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT REAL    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_058
INDEX IN is_customer_058
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 49420 ENDING AT 50286,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER59;
CREATE TABLE CUSTOMER59
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,

```

```

C_CREDIT    CHAR(2)    NOT NULL,
C_DISCOUNT REAL    NOT NULL,
C_DATA      VARCHAR(500) NOT NULL,
C_LAST      VARCHAR(16) NOT NULL,
C_FIRST     VARCHAR(16) NOT NULL,
C_STREET_1  VARCHAR(20) NOT NULL,
C_STREET_2  VARCHAR(20) NOT NULL,
C_CITY      VARCHAR(20) NOT NULL,
C_D_ID      SMALLINT   NOT NULL,
C_W_ID      INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE   DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER   NOT NULL
)
IN is_customer_059
INDEX IN is_customer_059
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 50287 ENDING AT 51153,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER60;
CREATE TABLE CUSTOMER60
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT REAL    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_060
INDEX IN is_customer_060
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 51154 ENDING AT 52020,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER61;
CREATE TABLE CUSTOMER61
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT REAL    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,

```

```

C_LAST      VARCHAR(16) NOT NULL,
C_FIRST     VARCHAR(16) NOT NULL,
C_STREET_1  VARCHAR(20) NOT NULL,
C_STREET_2  VARCHAR(20) NOT NULL,
C_CITY      VARCHAR(20) NOT NULL,
C_D_ID      SMALLINT   NOT NULL,
C_W_ID      INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE   DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_061
INDEX IN is_customer_061
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 52021 ENDING AT 52887,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER62;
CREATE TABLE CUSTOMER62
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT REAL    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER    NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_062
INDEX IN is_customer_062
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 52888 ENDING AT 53754,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER63;
CREATE TABLE CUSTOMER63
(
  C_ID      INTEGER    NOT NULL,
  C_STATE   CHAR(2)    NOT NULL,
  C_ZIP     CHAR(9)    NOT NULL,
  C_PHONE   CHAR(16)   NOT NULL,
  C_SINCE   TIMESTAMP  NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE  CHAR(2)    NOT NULL,
  C_CREDIT  CHAR(2)    NOT NULL,
  C_DISCOUNT REAL    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,

```

```

C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_063
INDEX IN is_customer_063
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 53755 ENDING AT 54621,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER64;
CREATE TABLE CUSTOMER64
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_064
INDEX IN is_customer_064
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 54622 ENDING AT 55488,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER65;
CREATE TABLE CUSTOMER65
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_065
INDEX IN is_customer_065
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 55489 ENDING AT 56355,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER66;
CREATE TABLE CUSTOMER66
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_066
INDEX IN is_customer_066
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 56356 ENDING AT 57222,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER67;
CREATE TABLE CUSTOMER67
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,

```

```

C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
)
IN ts_customer_067
INDEX IN is_customer_067
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 57223 ENDING AT 58089,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER68;
CREATE TABLE CUSTOMER68
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
)
IN ts_customer_068
INDEX IN is_customer_068
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 58090 ENDING AT 58956,
C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER69;
CREATE TABLE CUSTOMER69
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
)

```

```

IN ts_customer_069
INDEX IN is_customer_069
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 58957 ENDING AT 59823,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER70;
CREATE TABLE CUSTOMER70
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_070
INDEX IN is_customer_070
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 59824 ENDING AT 60690,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER71;
CREATE TABLE CUSTOMER71
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_071
INDEX IN is_customer_071
ORGANIZE BY KEY SEQUENCE (

```

```

C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 60691 ENDING AT 61557,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER72;
CREATE TABLE CUSTOMER72
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_072
INDEX IN is_customer_072
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 61558 ENDING AT 62424,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER73;
CREATE TABLE CUSTOMER73
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_073
INDEX IN is_customer_073
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 62425 ENDING AT 63291,
C_D_ID STARTING FROM 1 ENDING AT 10

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER74;
CREATE TABLE CUSTOMER74
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_074
INDEX IN is_customer_074
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 63292 ENDING AT 64158,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER75;
CREATE TABLE CUSTOMER75
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE TIMESTAMP NOT NULL,
C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT REAL NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE DECIMAL(12,2) NOT NULL,
C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_075
INDEX IN is_customer_075
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 64159 ENDING AT 65025,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER76;
CREATE TABLE CUSTOMER76
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_076
INDEX IN is_customer_076
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 65026 ENDING AT 65892,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER77;
CREATE TABLE CUSTOMER77
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_077
INDEX IN is_customer_077
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 65893 ENDING AT 66759,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER78;
CREATE TABLE CUSTOMER78

```

```

(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_078
INDEX IN is_customer_078
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 66760 ENDING AT 67626,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER79;
CREATE TABLE CUSTOMER79
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,
  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_079
INDEX IN is_customer_079
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 67627 ENDING AT 68493,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER80;
CREATE TABLE CUSTOMER80
(
  C_ID          INTEGER      NOT NULL,
  C_STATE      CHAR(2)      NOT NULL,

```

```

  C_ZIP        CHAR(9)      NOT NULL,
  C_PHONE      CHAR(16)     NOT NULL,
  C_SINCE      TIMESTAMP    NOT NULL,
  C_CREDIT_LIM DECIMAL(12,2) NOT NULL,
  C_MIDDLE     CHAR(2)      NOT NULL,
  C_CREDIT     CHAR(2)      NOT NULL,
  C_DISCOUNT  REAL         NOT NULL,
  C_DATA       VARCHAR(500) NOT NULL,
  C_LAST       VARCHAR(16)  NOT NULL,
  C_FIRST      VARCHAR(16)  NOT NULL,
  C_STREET_1   VARCHAR(20)  NOT NULL,
  C_STREET_2   VARCHAR(20)  NOT NULL,
  C_CITY       VARCHAR(20)  NOT NULL,
  C_D_ID       SMALLINT     NOT NULL,
  C_W_ID       INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER    NOT NULL,
  C_BALANCE    DECIMAL(12,2) NOT NULL,
  C_YTD_PAYMENT DECIMAL(12,2) NOT NULL,
  C_PAYMENT_CNT INTEGER     NOT NULL
)
IN ts_customer_080
INDEX IN is_customer_080
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 68494 ENDING AT 69360,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER81;
CREATE TABLE CUSTOMER81
(
  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_081
INDEX IN is_customer_081
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 69361 ENDING AT 70227,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER82;
CREATE TABLE CUSTOMER82
(
  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_082
INDEX IN is_customer_082
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 70228 ENDING AT 71094,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER83;
CREATE TABLE CUSTOMER83
(
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_083
INDEX IN is_customer_083
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 71095 ENDING AT 71961,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER84;
CREATE TABLE CUSTOMER84
(
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_084
INDEX IN is_customer_084
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 71962 ENDING AT 72828,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER85;
CREATE TABLE CUSTOMER85
(
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_085
INDEX IN is_customer_085
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 72829 ENDING AT 73695,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER86;
CREATE TABLE CUSTOMER86
(
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_086
INDEX IN is_customer_086
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 73696 ENDING AT 74562,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER87;
CREATE TABLE CUSTOMER87
(
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_087
INDEX IN is_customer_087
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 74563 ENDING AT 75429,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER88;
CREATE TABLE CUSTOMER88
(
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_088
INDEX IN is_customer_088
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 75430 ENDING AT 76296,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER89;
CREATE TABLE CUSTOMER89
(
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_089
INDEX IN is_customer_089
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 76297 ENDING AT 77163,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER90;
CREATE TABLE CUSTOMER90
(
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
)

```

```

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_090
INDEX IN is_customer_090
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 77164 ENDING AT 78030,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER91;
CREATE TABLE CUSTOMER91
(
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_091
INDEX IN is_customer_091
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 78031 ENDING AT 78897,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER92;
CREATE TABLE CUSTOMER92
(
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
)

```

```

C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_092
INDEX IN is_customer_092
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 78898 ENDING AT 79764,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER93;
CREATE TABLE CUSTOMER93
(
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_093
INDEX IN is_customer_093
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 79765 ENDING AT 80631,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER94;
CREATE TABLE CUSTOMER94
(
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_094

```

```

INDEX IN is_customer_094
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 80632 ENDING AT 81498,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER95;
CREATE TABLE CUSTOMER95
(
  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_095
INDEX IN is_customer_095
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 81499 ENDING AT 82365,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER96;
CREATE TABLE CUSTOMER96
(
  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_096
INDEX IN is_customer_096
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,

```

```

  C_W_ID STARTING FROM 82366 ENDING AT 83232,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER97;
CREATE TABLE CUSTOMER97
(
  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_097
INDEX IN is_customer_097
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 83233 ENDING AT 84099,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER98;
CREATE TABLE CUSTOMER98
(
  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_098
INDEX IN is_customer_098
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 84100 ENDING AT 84966,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER99;
CREATE TABLE CUSTOMER99
(
  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_099
INDEX IN is_customer_099
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 84967 ENDING AT 85833,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER100;
CREATE TABLE CUSTOMER100
(
  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_100
INDEX IN is_customer_100
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 85834 ENDING AT 86700,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```



```

DROP TABLE CUSTOMER101;
CREATE TABLE CUSTOMER101
(
  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_101
INDEX IN is_customer_101
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 86701 ENDING AT 87567,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER102;
CREATE TABLE CUSTOMER102
(
  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_102
INDEX IN is_customer_102
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 87568 ENDING AT 88434,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER103;
CREATE TABLE CUSTOMER103
(

```

```

  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_103
INDEX IN is_customer_103
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 88435 ENDING AT 89301,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER104;
CREATE TABLE CUSTOMER104
(
  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_104
INDEX IN is_customer_104
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 89302 ENDING AT 90168,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER105;
CREATE TABLE CUSTOMER105
(
  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_105
INDEX IN is_customer_105
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 90169 ENDING AT 91035,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER106;
CREATE TABLE CUSTOMER106
(
  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_106
INDEX IN is_customer_106
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 91036 ENDING AT 91902,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER107;
CREATE TABLE CUSTOMER107
(
  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_107
INDEX IN is_customer_107
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 91903 ENDING AT 92769,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER108;
CREATE TABLE CUSTOMER108
(
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_108
INDEX IN is_customer_108
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 92770 ENDING AT 93636,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER109;
CREATE TABLE CUSTOMER109
(
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_109
INDEX IN is_customer_109
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 93637 ENDING AT 94503,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER110;
CREATE TABLE CUSTOMER110
(
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_110
INDEX IN is_customer_110
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 94504 ENDING AT 95370,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER111;
CREATE TABLE CUSTOMER111
(
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_111
INDEX IN is_customer_111
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 95371 ENDING AT 96237,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER112;
CREATE TABLE CUSTOMER112
(
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_112
INDEX IN is_customer_112
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 96238 ENDING AT 97104,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER113;
CREATE TABLE CUSTOMER113
(
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_113
INDEX IN is_customer_113
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 97105 ENDING AT 97971,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER114;
CREATE TABLE CUSTOMER114
(
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_114
INDEX IN is_customer_114
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 97972 ENDING AT 98838,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER115;
CREATE TABLE CUSTOMER115
(
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_115
INDEX IN is_customer_115
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 98839 ENDING AT 99705,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER116;
CREATE TABLE CUSTOMER116
(
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_116
INDEX IN is_customer_116
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 99706 ENDING AT 100572,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER117;
CREATE TABLE CUSTOMER117
(
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_117
INDEX IN is_customer_117
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 100573 ENDING AT 101439,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER118;
CREATE TABLE CUSTOMER118
(
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_118
INDEX IN is_customer_118
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 101440 ENDING AT 102306,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER119;
CREATE TABLE CUSTOMER119
(
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_119
INDEX IN is_customer_119

```

```

ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 102307 ENDING AT 103173,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER120;
CREATE TABLE CUSTOMER120

```

```

(
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_120
INDEX IN is_customer_120
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 103174 ENDING AT 104040,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

### DDL/CRTB DISTRICT.ddl

```

connect to TPCC in share mode;
DROP TABLE DISTRICT1;
CREATE TABLE DISTRICT1
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_001
INDEX IN ts_dist_001
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 1 ENDING AT 867
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT2;

```

```

CREATE TABLE DISTRICT2
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_002
INDEX IN ts_dist_002
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 868 ENDING AT 1734
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT3;
CREATE TABLE DISTRICT3
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_003
INDEX IN ts_dist_003
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 1735 ENDING AT 2601
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT4;
CREATE TABLE DISTRICT4
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_004
INDEX IN ts_dist_004
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 2602 ENDING AT 3468
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT5;
CREATE TABLE DISTRICT5

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_005
INDEX IN ts_dist_005
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 3469 ENDING AT 4335
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT6;
CREATE TABLE DISTRICT6
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_006
INDEX IN ts_dist_006
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 4336 ENDING AT 5202
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT7;
CREATE TABLE DISTRICT7
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_007
INDEX IN ts_dist_007
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 5203 ENDING AT 6069
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT8;
CREATE TABLE DISTRICT8
(

```

```

D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_008
INDEX IN ts_dist_008
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 6070 ENDING AT 6936
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT9;
CREATE TABLE DISTRICT9
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_009
INDEX IN ts_dist_009
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 6937 ENDING AT 7803
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT10;
CREATE TABLE DISTRICT10
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_010
INDEX IN ts_dist_010
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 7804 ENDING AT 8670
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT11;
CREATE TABLE DISTRICT11
(
D_NEXT_O_ID INTEGER NOT NULL,

```

```

D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_011
INDEX IN ts_dist_011
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 8671 ENDING AT 9537
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT12;
CREATE TABLE DISTRICT12
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_012
INDEX IN ts_dist_012
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 9538 ENDING AT 10404
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT13;
CREATE TABLE DISTRICT13
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_013
INDEX IN ts_dist_013
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 10405 ENDING AT 11271
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT14;
CREATE TABLE DISTRICT14
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,

```

```

D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_014
INDEX IN ts_dist_014
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 11272 ENDING AT 12138
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT15;
CREATE TABLE DISTRICT15
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_015
INDEX IN ts_dist_015
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 12139 ENDING AT 13005
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT16;
CREATE TABLE DISTRICT16
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_016
INDEX IN ts_dist_016
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 13006 ENDING AT 13872
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT17;
CREATE TABLE DISTRICT17
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,

```

```

D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_017
INDEX IN ts_dist_017
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 13873 ENDING AT 14739
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT18;
CREATE TABLE DISTRICT18
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_018
INDEX IN ts_dist_018
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 14740 ENDING AT 15606
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT19;
CREATE TABLE DISTRICT19
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_019
INDEX IN ts_dist_019
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 15607 ENDING AT 16473
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT20;
CREATE TABLE DISTRICT20
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,

```

```

D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
)
IN ts_dist_020
INDEX IN ts_dist_020
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 16474 ENDING AT 17340
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT21;
CREATE TABLE DISTRICT21
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
)
IN ts_dist_021
INDEX IN ts_dist_021
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 17341 ENDING AT 18207
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT22;
CREATE TABLE DISTRICT22
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
)
IN ts_dist_022
INDEX IN ts_dist_022
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 18208 ENDING AT 19074
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT23;
CREATE TABLE DISTRICT23
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,

```

```

D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
)
IN ts_dist_023
INDEX IN ts_dist_023
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 19075 ENDING AT 19941
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT24;
CREATE TABLE DISTRICT24
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
)
IN ts_dist_024
INDEX IN ts_dist_024
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 19942 ENDING AT 20808
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT25;
CREATE TABLE DISTRICT25
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
)
IN ts_dist_025
INDEX IN ts_dist_025
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 20809 ENDING AT 21675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT26;
CREATE TABLE DISTRICT26
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,

```

```

D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_026
INDEX IN ts_dist_026
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 21676 ENDING AT 22542
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT27;
CREATE TABLE DISTRICT27
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_027
INDEX IN ts_dist_027
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 22543 ENDING AT 23409
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT28;
CREATE TABLE DISTRICT28
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_028
INDEX IN ts_dist_028
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 23410 ENDING AT 24276
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT29;
CREATE TABLE DISTRICT29
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,

```

```

D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_029
INDEX IN ts_dist_029
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 24277 ENDING AT 25143
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT30;
CREATE TABLE DISTRICT30
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_030
INDEX IN ts_dist_030
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 25144 ENDING AT 26010
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT31;
CREATE TABLE DISTRICT31
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_031
INDEX IN ts_dist_031
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 26011 ENDING AT 26877
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT32;
CREATE TABLE DISTRICT32
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,

```

```

D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_032
INDEX IN ts_dist_032
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 26878 ENDING AT 27744
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT33;
CREATE TABLE DISTRICT33
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_033
INDEX IN ts_dist_033
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 27745 ENDING AT 28611
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT34;
CREATE TABLE DISTRICT34
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_034
INDEX IN ts_dist_034
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 28612 ENDING AT 29478
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT35;
CREATE TABLE DISTRICT35
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,

```

```

D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_035
INDEX IN ts_dist_035
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 29479 ENDING AT 30345
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT36;
CREATE TABLE DISTRICT36
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_036
INDEX IN ts_dist_036
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 30346 ENDING AT 31212
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT37;
CREATE TABLE DISTRICT37
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_037
INDEX IN ts_dist_037
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 31213 ENDING AT 32079
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT38;
CREATE TABLE DISTRICT38
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL,

```

```

D_W_ID INTEGER NOT NULL
)
IN ts_dist_038
INDEX IN ts_dist_038
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 32080 ENDING AT 32946
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT39;
CREATE TABLE DISTRICT39
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_039
INDEX IN ts_dist_039
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 32947 ENDING AT 33813
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT40;
CREATE TABLE DISTRICT40
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_040
INDEX IN ts_dist_040
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 33814 ENDING AT 34680
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT41;
CREATE TABLE DISTRICT41
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL,

```

```

)
IN ts_dist_041
INDEX IN ts_dist_041
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 34681 ENDING AT 35547
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT42;
CREATE TABLE DISTRICT42
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_042
INDEX IN ts_dist_042
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 35548 ENDING AT 36414
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT43;
CREATE TABLE DISTRICT43
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_043
INDEX IN ts_dist_043
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 36415 ENDING AT 37281
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT44;
CREATE TABLE DISTRICT44
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL,
)

```



```

IN ts_dist_044
INDEX IN ts_dist_044
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 37282 ENDING AT 38148
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT45;
CREATE TABLE DISTRICT45
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_045
INDEX IN ts_dist_045
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 38149 ENDING AT 39015
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT46;
CREATE TABLE DISTRICT46
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_046
INDEX IN ts_dist_046
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 39016 ENDING AT 39882
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT47;
CREATE TABLE DISTRICT47
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_047

```

```

INDEX IN ts_dist_047
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 39883 ENDING AT 40749
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT48;
CREATE TABLE DISTRICT48
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_048
INDEX IN ts_dist_048
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 40750 ENDING AT 41616
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT49;
CREATE TABLE DISTRICT49
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_049
INDEX IN ts_dist_049
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 41617 ENDING AT 42483
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT50;
CREATE TABLE DISTRICT50
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_050
INDEX IN ts_dist_050

```

```

ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 42484 ENDING AT 43350
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT51;
CREATE TABLE DISTRICT51
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_051
INDEX IN ts_dist_051
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 43351 ENDING AT 44217
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT52;
CREATE TABLE DISTRICT52
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_052
INDEX IN ts_dist_052
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 44218 ENDING AT 45084
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT53;
CREATE TABLE DISTRICT53
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_053
INDEX IN ts_dist_053
ORGANIZE BY KEY SEQUENCE (

```

```

D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 45085 ENDING AT 45951
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT54;
CREATE TABLE DISTRICT54
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_054
INDEX IN ts_dist_054
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 45952 ENDING AT 46818
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT55;
CREATE TABLE DISTRICT55
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_055
INDEX IN ts_dist_055
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 46819 ENDING AT 47685
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT56;
CREATE TABLE DISTRICT56
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_056
INDEX IN ts_dist_056
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,

```

```

D_W_ID STARTING FROM 47686 ENDING AT 48552
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT57;
CREATE TABLE DISTRICT57
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_057
INDEX IN ts_dist_057
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 48553 ENDING AT 49419
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT58;
CREATE TABLE DISTRICT58
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_058
INDEX IN ts_dist_058
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 49420 ENDING AT 50286
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT59;
CREATE TABLE DISTRICT59
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_059
INDEX IN ts_dist_059
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 50287 ENDING AT 51153
)

```

```

)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT60;
CREATE TABLE DISTRICT60
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_060
INDEX IN ts_dist_060
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 51154 ENDING AT 52020
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT61;
CREATE TABLE DISTRICT61
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_061
INDEX IN ts_dist_061
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 52021 ENDING AT 52887
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT62;
CREATE TABLE DISTRICT62
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX REAL NOT NULL,
D_YTD DECIMAL(12,2) NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_STATE CHAR(2) NOT NULL,
D_ZIP CHAR(9) NOT NULL,
D_ID SMALLINT NOT NULL,
D_W_ID INTEGER NOT NULL
)
IN ts_dist_062
INDEX IN ts_dist_062
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 52888 ENDING AT 53754
)
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT63;
CREATE TABLE DISTRICT63
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_063
INDEX IN ts_dist_063
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 53755 ENDING AT 54621
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT64;
CREATE TABLE DISTRICT64
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_064
INDEX IN ts_dist_064
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 54622 ENDING AT 55488
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT65;
CREATE TABLE DISTRICT65
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_065
INDEX IN ts_dist_065
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 55489 ENDING AT 56355
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT66;
CREATE TABLE DISTRICT66
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_066
INDEX IN ts_dist_066
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 56356 ENDING AT 57222
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT67;
CREATE TABLE DISTRICT67
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_067
INDEX IN ts_dist_067
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 57223 ENDING AT 58089
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT68;
CREATE TABLE DISTRICT68
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_068
INDEX IN ts_dist_068
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 58090 ENDING AT 58956
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE DISTRICT69;
CREATE TABLE DISTRICT69
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_069
INDEX IN ts_dist_069
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 58957 ENDING AT 59823
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT70;
CREATE TABLE DISTRICT70
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_070
INDEX IN ts_dist_070
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 59824 ENDING AT 60690
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT71;
CREATE TABLE DISTRICT71
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_071
INDEX IN ts_dist_071
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 60691 ENDING AT 61557
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE DISTRICT72;
CREATE TABLE DISTRICT72
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_072
INDEX IN ts_dist_072
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 61558 ENDING AT 62424
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT73;
CREATE TABLE DISTRICT73
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_073
INDEX IN ts_dist_073
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 62425 ENDING AT 63291
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT74;
CREATE TABLE DISTRICT74
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_074
INDEX IN ts_dist_074
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 63292 ENDING AT 64158
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT75;

```

```

CREATE TABLE DISTRICT75
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_075
INDEX IN ts_dist_075
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 64159 ENDING AT 65025
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT76;
CREATE TABLE DISTRICT76
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_076
INDEX IN ts_dist_076
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 65026 ENDING AT 65892
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT77;
CREATE TABLE DISTRICT77
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_077
INDEX IN ts_dist_077
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 65893 ENDING AT 66759
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT78;
CREATE TABLE DISTRICT78

```

```

(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_078
INDEX IN ts_dist_078
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 66760 ENDING AT 67626
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT79;
CREATE TABLE DISTRICT79
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_079
INDEX IN ts_dist_079
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 67627 ENDING AT 68493
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT80;
CREATE TABLE DISTRICT80
(
  D_NEXT_O_ID INTEGER NOT NULL,
  D_TAX REAL NOT NULL,
  D_YTD DECIMAL(12,2) NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_STATE CHAR(2) NOT NULL,
  D_ZIP CHAR(9) NOT NULL,
  D_ID SMALLINT NOT NULL,
  D_W_ID INTEGER NOT NULL
)
IN ts_dist_080
INDEX IN ts_dist_080
ORGANIZE BY KEY SEQUENCE (
  D_ID STARTING FROM 1 ENDING AT 10,
  D_W_ID STARTING FROM 68494 ENDING AT 69360
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT81;
CREATE TABLE DISTRICT81
(

```

```

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_dist_081
INDEX IN ts_dist_081
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 69361 ENDING AT 70227
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT82;
CREATE TABLE DISTRICT82
(
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_dist_082
INDEX IN ts_dist_082
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 70228 ENDING AT 71094
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT83;
CREATE TABLE DISTRICT83
(
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_dist_083
INDEX IN ts_dist_083
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 71095 ENDING AT 71961
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT84;
CREATE TABLE DISTRICT84
(
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_dist_084
INDEX IN ts_dist_084
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 71962 ENDING AT 72828
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT85;
CREATE TABLE DISTRICT85
(
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_dist_085
INDEX IN ts_dist_085
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 72829 ENDING AT 73695
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT86;
CREATE TABLE DISTRICT86
(
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_dist_086
INDEX IN ts_dist_086
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 73696 ENDING AT 74562
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT87;
CREATE TABLE DISTRICT87
(
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_dist_087
INDEX IN ts_dist_087
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 74563 ENDING AT 75429
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT88;
CREATE TABLE DISTRICT88
(
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_dist_088
INDEX IN ts_dist_088
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 75430 ENDING AT 76296
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT89;
CREATE TABLE DISTRICT89
(
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_dist_089
INDEX IN ts_dist_089
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 76297 ENDING AT 77163
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT90;
CREATE TABLE DISTRICT90
(
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_dist_090
INDEX IN ts_dist_090
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 77164 ENDING AT 78030
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT91;
CREATE TABLE DISTRICT91
(
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_dist_091
INDEX IN ts_dist_091
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 78031 ENDING AT 78897
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT92;
CREATE TABLE DISTRICT92
(
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_dist_092
INDEX IN ts_dist_092
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 78898 ENDING AT 79764
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT93;
CREATE TABLE DISTRICT93
(
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_dist_093
INDEX IN ts_dist_093
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 79765 ENDING AT 80631
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT94;
CREATE TABLE DISTRICT94
(
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_dist_094
INDEX IN ts_dist_094
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 80632 ENDING AT 81498
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT95;
CREATE TABLE DISTRICT95
(
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_dist_095
INDEX IN ts_dist_095
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 81499 ENDING AT 82365
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT96;
CREATE TABLE DISTRICT96
(
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_dist_096
INDEX IN ts_dist_096
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 82366 ENDING AT 83232
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT97;
CREATE TABLE DISTRICT97
(
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_dist_097
INDEX IN ts_dist_097
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 83233 ENDING AT 84099
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT98;
CREATE TABLE DISTRICT98
(
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_dist_098
INDEX IN ts_dist_098
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 84100 ENDING AT 84966
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT99;
CREATE TABLE DISTRICT99
(
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_dist_099
INDEX IN ts_dist_099
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 84967 ENDING AT 85833
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT100;
CREATE TABLE DISTRICT100
(
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_dist_100
INDEX IN ts_dist_100
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 85834 ENDING AT 86700
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT101;
CREATE TABLE DISTRICT101
(
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_dist_101
INDEX IN ts_dist_101
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 86701 ENDING AT 87567
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT102;
CREATE TABLE DISTRICT102
(
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_dist_102
INDEX IN ts_dist_102
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 87568 ENDING AT 88434
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT103;
CREATE TABLE DISTRICT103
(
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_dist_103
INDEX IN ts_dist_103
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 88435 ENDING AT 89301
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT104;
CREATE TABLE DISTRICT104
(
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_dist_104
INDEX IN ts_dist_104
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 89302 ENDING AT 90168
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT105;
CREATE TABLE DISTRICT105
(
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_dist_105
INDEX IN ts_dist_105
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 90169 ENDING AT 91035
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT106;
CREATE TABLE DISTRICT106
(
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_dist_106
INDEX IN ts_dist_106
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 91036 ENDING AT 91902
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT107;
CREATE TABLE DISTRICT107
(
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_dist_107
INDEX IN ts_dist_107
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 91903 ENDING AT 92769
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT108;
CREATE TABLE DISTRICT108
(
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_dist_108
INDEX IN ts_dist_108
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 92770 ENDING AT 93636
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT109;
CREATE TABLE DISTRICT109
(
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_dist_109
INDEX IN ts_dist_109
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 93637 ENDING AT 94503
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT110;
CREATE TABLE DISTRICT110
(
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_dist_110
INDEX IN ts_dist_110
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 94504 ENDING AT 95370
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT111;
CREATE TABLE DISTRICT111
(
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,
)

```

```

D_W_ID INTEGER NOT NULL
)
IN ts_dist_111
INDEX IN ts_dist_111
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 95371 ENDING AT 96237
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT112;
CREATE TABLE DISTRICT112
(
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_dist_112
INDEX IN ts_dist_112
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 96238 ENDING AT 97104
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT113;
CREATE TABLE DISTRICT113
(
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_dist_113
INDEX IN ts_dist_113
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 97105 ENDING AT 97971
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT114;
CREATE TABLE DISTRICT114
(
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_dist_114
INDEX IN ts_dist_114
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 97972 ENDING AT 98838
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT115;
CREATE TABLE DISTRICT115
(
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_dist_115
INDEX IN ts_dist_115
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 98839 ENDING AT 99705
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT116;
CREATE TABLE DISTRICT116
(
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_dist_116
INDEX IN ts_dist_116
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 99706 ENDING AT 100572
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT117;
CREATE TABLE DISTRICT117
(
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_dist_117
INDEX IN ts_dist_117
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 100573 ENDING AT 101439
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT118;
CREATE TABLE DISTRICT118
(
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_dist_118
INDEX IN ts_dist_118
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 101440 ENDING AT 102306
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT119;
CREATE TABLE DISTRICT119
(
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_dist_119
INDEX IN ts_dist_119
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 102307 ENDING AT 103173
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT120;
CREATE TABLE DISTRICT120
(
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_dist_120

```

```

INDEX IN ts_dist_120
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 103174 ENDING AT 104040
)
)
ALLOW OVERFLOW;
connect reset;

```

### DDL/CRTB HISTORY.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;
connect to TPCC in share mode;
DROP TABLE HISTORY2;
CREATE TABLE HISTORY2
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE TIMESTAMP NOT NULL,
H_AMOUNT DECIMAL(6,2) NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_002
INDEX IN ts_history_002;
ALTER TABLE HISTORY2 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY3;
CREATE TABLE HISTORY3
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE TIMESTAMP NOT NULL,
H_AMOUNT DECIMAL(6,2) NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_003
INDEX IN ts_history_003;
ALTER TABLE HISTORY3 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY4;
CREATE TABLE HISTORY4
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE TIMESTAMP NOT NULL,

```

```

H_AMOUNT DECIMAL(6,2) NOT NULL,
H_DATA CHAR(24) NOT NULL
)
)
IN ts_history_004
INDEX IN ts_history_004;
ALTER TABLE HISTORY4 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY5;
CREATE TABLE HISTORY5
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE TIMESTAMP NOT NULL,
H_AMOUNT DECIMAL(6,2) NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_005
INDEX IN ts_history_005;
ALTER TABLE HISTORY5 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY6;
CREATE TABLE HISTORY6
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE TIMESTAMP NOT NULL,
H_AMOUNT DECIMAL(6,2) NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_006
INDEX IN ts_history_006;
ALTER TABLE HISTORY6 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY7;
CREATE TABLE HISTORY7
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE TIMESTAMP NOT NULL,
H_AMOUNT DECIMAL(6,2) NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_007
INDEX IN ts_history_007;
ALTER TABLE HISTORY7 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY8;
CREATE TABLE HISTORY8
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE TIMESTAMP NOT NULL,
H_AMOUNT DECIMAL(6,2) NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_008
INDEX IN ts_history_008;

```



















```

connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY112;
CREATE TABLE HISTORY112
(
  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_112
INDEX IN ts_history_112;
ALTER TABLE HISTORY112 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY113;
CREATE TABLE HISTORY113
(
  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_113
INDEX IN ts_history_113;
ALTER TABLE HISTORY113 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY114;
CREATE TABLE HISTORY114
(
  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_114
INDEX IN ts_history_114;
ALTER TABLE HISTORY114 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY115;
CREATE TABLE HISTORY115
(
  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_115
INDEX IN ts_history_115;
ALTER TABLE HISTORY115 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY116;
CREATE TABLE HISTORY116
(
  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_116
INDEX IN ts_history_116;
ALTER TABLE HISTORY116 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY117;
CREATE TABLE HISTORY117
(
  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_117
INDEX IN ts_history_117;
ALTER TABLE HISTORY117 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY118;
CREATE TABLE HISTORY118
(
  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_118
INDEX IN ts_history_118;
ALTER TABLE HISTORY118 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY119;
CREATE TABLE HISTORY119
(
  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_119
INDEX IN ts_history_119;
ALTER TABLE HISTORY119 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY120;
CREATE TABLE HISTORY120
(
  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_120
INDEX IN ts_history_120;
ALTER TABLE HISTORY120 APPEND ON;
connect reset;

```

```

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_116
INDEX IN ts_history_116;
ALTER TABLE HISTORY116 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY117;
CREATE TABLE HISTORY117
(
  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_117
INDEX IN ts_history_117;
ALTER TABLE HISTORY117 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY118;
CREATE TABLE HISTORY118
(
  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_118
INDEX IN ts_history_118;
ALTER TABLE HISTORY118 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY119;
CREATE TABLE HISTORY119
(
  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_119
INDEX IN ts_history_119;
ALTER TABLE HISTORY119 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY120;
CREATE TABLE HISTORY120
(
  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_120
INDEX IN ts_history_120;
ALTER TABLE HISTORY120 APPEND ON;
connect reset;

```

```

H_DATE    TIMESTAMP NOT NULL,
H_AMOUNT  DECIMAL(6,2) NOT NULL,
H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_120
INDEX IN ts_history_120;
ALTER TABLE HISTORY120 APPEND ON;
connect reset;

```

### DDL/CRTB ITEM.ddl

```

connect to TPCC in share mode;
DROP TABLE ITEM;
CREATE TABLE ITEM
(
  I_NAME   CHAR(24)  NOT NULL,
  I_PRICE  DECIMAL(5,2) NOT NULL,
  I_DATA   VARCHAR(50) NOT NULL,
  I_IM_ID  INTEGER   NOT NULL,
  I_ID     INTEGER   NOT NULL
)
IN ts_item_001
INDEX IN ts_item_001
ORGANIZE BY KEY SEQUENCE (
  I_ID STARTING FROM 1 ENDING AT 10000
)
ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;
connect reset;

```

### DDL/CRTB NEW ORDERA.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA1;
CREATE TABLE NEW_ORDERA1
(
  NO_O_ID  INTEGER   NOT NULL,
  NO_D_ID  SMALLINT  NOT NULL,
  NO_W_ID  INTEGER   NOT NULL
)
IN ts_newordA_001
INDEX IN ts_newordA_001
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 1 ENDING AT 867,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA2;
CREATE TABLE NEW_ORDERA2
(
  NO_O_ID  INTEGER   NOT NULL,
  NO_D_ID  SMALLINT  NOT NULL,
  NO_W_ID  INTEGER   NOT NULL
)
IN ts_newordA_002
INDEX IN ts_newordA_002
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 868 ENDING AT 1734,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA3;
CREATE TABLE NEW_ORDERA3
(

```

```

NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_003
INDEX IN ts_newordA_003
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 1735 ENDING AT 2601,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA4;
CREATE TABLE NEW_ORDERA4
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_004
INDEX IN ts_newordA_004
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 2602 ENDING AT 3468,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA5;
CREATE TABLE NEW_ORDERA5
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_005
INDEX IN ts_newordA_005
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 3469 ENDING AT 4335,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA6;
CREATE TABLE NEW_ORDERA6
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_006
INDEX IN ts_newordA_006
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 4336 ENDING AT 5202,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA7;
CREATE TABLE NEW_ORDERA7
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordA_007

```

```

INDEX IN ts_newordA_007
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 5203 ENDING AT 6069,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA8;
CREATE TABLE NEW_ORDERA8
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_008
INDEX IN ts_newordA_008
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 6070 ENDING AT 6936,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA9;
CREATE TABLE NEW_ORDERA9
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_009
INDEX IN ts_newordA_009
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 6937 ENDING AT 7803,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA10;
CREATE TABLE NEW_ORDERA10
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_010
INDEX IN ts_newordA_010
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 7804 ENDING AT 8670,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA11;
CREATE TABLE NEW_ORDERA11
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_011
INDEX IN ts_newordA_011
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 8671 ENDING AT 9537,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA12;
CREATE TABLE NEW_ORDERA12
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_012
INDEX IN ts_newordA_012
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 9538 ENDING AT 10404,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA13;
CREATE TABLE NEW_ORDERA13
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_013
INDEX IN ts_newordA_013
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 10405 ENDING AT 11271,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA14;
CREATE TABLE NEW_ORDERA14
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_014
INDEX IN ts_newordA_014
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 11272 ENDING AT 12138,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA15;
CREATE TABLE NEW_ORDERA15
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordA_015
INDEX IN ts_newordA_015
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 12139 ENDING AT 13005,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA16;

```

```

CREATE TABLE NEW_ORDERA16
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_016
INDEX IN ts_newordA_016
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 13006 ENDING AT 13872,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA17;
CREATE TABLE NEW_ORDERA17

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_017
INDEX IN ts_newordA_017
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 13873 ENDING AT 14739,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA18;
CREATE TABLE NEW_ORDERA18

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_018
INDEX IN ts_newordA_018
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 14740 ENDING AT 15606,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA19;
CREATE TABLE NEW_ORDERA19

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_019
INDEX IN ts_newordA_019
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 15607 ENDING AT 16473,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA20;
CREATE TABLE NEW_ORDERA20

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

)
IN ts_newordA_020
INDEX IN ts_newordA_020
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 16474 ENDING AT 17340,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA21;
CREATE TABLE NEW_ORDERA21

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_021
INDEX IN ts_newordA_021
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 17341 ENDING AT 18207,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA22;
CREATE TABLE NEW_ORDERA22

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_022
INDEX IN ts_newordA_022
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 18208 ENDING AT 19074,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA23;
CREATE TABLE NEW_ORDERA23

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_023
INDEX IN ts_newordA_023
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 19075 ENDING AT 19941,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA24;
CREATE TABLE NEW_ORDERA24

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_024
INDEX IN ts_newordA_024
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 19942 ENDING AT 20808,

```

```

  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA25;
CREATE TABLE NEW_ORDERA25

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_025
INDEX IN ts_newordA_025
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 20809 ENDING AT 21675,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA26;
CREATE TABLE NEW_ORDERA26

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_026
INDEX IN ts_newordA_026
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 21676 ENDING AT 22542,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA27;
CREATE TABLE NEW_ORDERA27

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_027
INDEX IN ts_newordA_027
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 22543 ENDING AT 23409,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA28;
CREATE TABLE NEW_ORDERA28

```

```

(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)

```

```

IN ts_newordA_028
INDEX IN ts_newordA_028
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 23410 ENDING AT 24276,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

ALLOW OVERFLOW;

```

connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA29;
CREATE TABLE NEW_ORDERA29
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_029
INDEX IN ts_newordA_029
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 24277 ENDING AT 25143,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA30;
CREATE TABLE NEW_ORDERA30
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_030
INDEX IN ts_newordA_030
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 25144 ENDING AT 26010,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA31;
CREATE TABLE NEW_ORDERA31
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_031
INDEX IN ts_newordA_031
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 26011 ENDING AT 26877,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA32;
CREATE TABLE NEW_ORDERA32
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_032
INDEX IN ts_newordA_032
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 26878 ENDING AT 27744,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA33;
CREATE TABLE NEW_ORDERA33
(
  NO_O_ID    INTEGER NOT NULL,

```

```

  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_033
INDEX IN ts_newordA_033
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 27745 ENDING AT 28611,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA34;
CREATE TABLE NEW_ORDERA34
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_034
INDEX IN ts_newordA_034
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 28612 ENDING AT 29478,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA35;
CREATE TABLE NEW_ORDERA35
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_035
INDEX IN ts_newordA_035
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 29479 ENDING AT 30345,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA36;
CREATE TABLE NEW_ORDERA36
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_036
INDEX IN ts_newordA_036
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 30346 ENDING AT 31212,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA37;
CREATE TABLE NEW_ORDERA37
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_037
INDEX IN ts_newordA_037

```

```

ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 31213 ENDING AT 32079,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA38;
CREATE TABLE NEW_ORDERA38
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_038
INDEX IN ts_newordA_038
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 32080 ENDING AT 32946,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA39;
CREATE TABLE NEW_ORDERA39
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_039
INDEX IN ts_newordA_039
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 32947 ENDING AT 33813,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA40;
CREATE TABLE NEW_ORDERA40
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_040
INDEX IN ts_newordA_040
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 33814 ENDING AT 34680,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA41;
CREATE TABLE NEW_ORDERA41
(
  NO_O_ID    INTEGER NOT NULL,
  NO_D_ID    SMALLINT NOT NULL,
  NO_W_ID    INTEGER NOT NULL
)
IN ts_newordA_041
INDEX IN ts_newordA_041
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 34681 ENDING AT 35547,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA42;
CREATE TABLE NEW_ORDERA42
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_042
INDEX IN ts_newordA_042
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 35548 ENDING AT 36414,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA43;
CREATE TABLE NEW_ORDERA43
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_043
INDEX IN ts_newordA_043
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 36415 ENDING AT 37281,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA44;
CREATE TABLE NEW_ORDERA44
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_044
INDEX IN ts_newordA_044
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 37282 ENDING AT 38148,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA45;
CREATE TABLE NEW_ORDERA45
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_045
INDEX IN ts_newordA_045
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 38149 ENDING AT 39015,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA46;
CREATE TABLE NEW_ORDERA46

```

```

(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_046
INDEX IN ts_newordA_046
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 39016 ENDING AT 39882,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA47;
CREATE TABLE NEW_ORDERA47
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_047
INDEX IN ts_newordA_047
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 39883 ENDING AT 40749,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA48;
CREATE TABLE NEW_ORDERA48
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_048
INDEX IN ts_newordA_048
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 40750 ENDING AT 41616,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA49;
CREATE TABLE NEW_ORDERA49
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_049
INDEX IN ts_newordA_049
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 41617 ENDING AT 42483,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA50;
CREATE TABLE NEW_ORDERA50
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)

```

```

IN ts_newordA_050
INDEX IN ts_newordA_050
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 42484 ENDING AT 43350,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA51;
CREATE TABLE NEW_ORDERA51
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_051
INDEX IN ts_newordA_051
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 43351 ENDING AT 44217,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA52;
CREATE TABLE NEW_ORDERA52
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_052
INDEX IN ts_newordA_052
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 44218 ENDING AT 45084,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA53;
CREATE TABLE NEW_ORDERA53
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_053
INDEX IN ts_newordA_053
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 45085 ENDING AT 45951,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA54;
CREATE TABLE NEW_ORDERA54
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_newordA_054
INDEX IN ts_newordA_054
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 45952 ENDING AT 46818,
  NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA55;
CREATE TABLE NEW_ORDERA55
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_055
INDEX IN ts_newordA_055
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 46819 ENDING AT 47685,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA56;
CREATE TABLE NEW_ORDERA56
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_056
INDEX IN ts_newordA_056
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 47686 ENDING AT 48552,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA57;
CREATE TABLE NEW_ORDERA57
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_057
INDEX IN ts_newordA_057
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 48553 ENDING AT 49419,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA58;
CREATE TABLE NEW_ORDERA58
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_058
INDEX IN ts_newordA_058
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 49420 ENDING AT 50286,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE NEW_ORDERA59;
CREATE TABLE NEW_ORDERA59
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_059
INDEX IN ts_newordA_059
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 50287 ENDING AT 51153,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA60;
CREATE TABLE NEW_ORDERA60
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_060
INDEX IN ts_newordA_060
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 51154 ENDING AT 52020,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA61;
CREATE TABLE NEW_ORDERA61
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_061
INDEX IN ts_newordA_061
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 52021 ENDING AT 52887,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA62;
CREATE TABLE NEW_ORDERA62
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_062
INDEX IN ts_newordA_062
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 52888 ENDING AT 53754,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA63;
CREATE TABLE NEW_ORDERA63
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,

```

```

NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_063
INDEX IN ts_newordA_063
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 53755 ENDING AT 54621,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA64;
CREATE TABLE NEW_ORDERA64
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_064
INDEX IN ts_newordA_064
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 54622 ENDING AT 55488,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA65;
CREATE TABLE NEW_ORDERA65
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_065
INDEX IN ts_newordA_065
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 55489 ENDING AT 56355,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA66;
CREATE TABLE NEW_ORDERA66
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_066
INDEX IN ts_newordA_066
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 56356 ENDING AT 57222,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA67;
CREATE TABLE NEW_ORDERA67
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_067
INDEX IN ts_newordA_067
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 57223 ENDING AT 58089,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA68;
CREATE TABLE NEW_ORDERA68
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_068
INDEX IN ts_newordA_068
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 58090 ENDING AT 58956,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA69;
CREATE TABLE NEW_ORDERA69
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_069
INDEX IN ts_newordA_069
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 58957 ENDING AT 59823,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA70;
CREATE TABLE NEW_ORDERA70
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_070
INDEX IN ts_newordA_070
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 59824 ENDING AT 60690,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA71;
CREATE TABLE NEW_ORDERA71
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_071
INDEX IN ts_newordA_071
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 60691 ENDING AT 61557,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA72;
CREATE TABLE NEW_ORDERA72
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_072
INDEX IN ts_newordA_072
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 61558 ENDING AT 62424,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA73;
CREATE TABLE NEW_ORDERA73
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_073
INDEX IN ts_newordA_073
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 62425 ENDING AT 63291,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA74;
CREATE TABLE NEW_ORDERA74
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_074
INDEX IN ts_newordA_074
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 63292 ENDING AT 64158,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA75;
CREATE TABLE NEW_ORDERA75
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_075
INDEX IN ts_newordA_075
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 64159 ENDING AT 65025,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA76;
CREATE TABLE NEW_ORDERA76
(

```

```

NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_076
INDEX IN ts_newordA_076
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 65026 ENDING AT 65892,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA77;
CREATE TABLE NEW_ORDERA77
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_077
INDEX IN ts_newordA_077
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 65893 ENDING AT 66759,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA78;
CREATE TABLE NEW_ORDERA78
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_078
INDEX IN ts_newordA_078
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 66760 ENDING AT 67626,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA79;
CREATE TABLE NEW_ORDERA79
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_079
INDEX IN ts_newordA_079
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 67627 ENDING AT 68493,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA80;
CREATE TABLE NEW_ORDERA80
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_080

```



```

INDEX IN ts_newordA_080
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 68494 ENDING AT 69360,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA81;
CREATE TABLE NEW_ORDERA81
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_081
INDEX IN ts_newordA_081
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 69361 ENDING AT 70227,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA82;
CREATE TABLE NEW_ORDERA82
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_082
INDEX IN ts_newordA_082
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 70228 ENDING AT 71094,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA83;
CREATE TABLE NEW_ORDERA83
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_083
INDEX IN ts_newordA_083
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 71095 ENDING AT 71961,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA84;
CREATE TABLE NEW_ORDERA84
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_084
INDEX IN ts_newordA_084
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 71962 ENDING AT 72828,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)

```

```

)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA85;
CREATE TABLE NEW_ORDERA85
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_085
INDEX IN ts_newordA_085
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 72829 ENDING AT 73695,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA86;
CREATE TABLE NEW_ORDERA86
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_086
INDEX IN ts_newordA_086
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 73696 ENDING AT 74562,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA87;
CREATE TABLE NEW_ORDERA87
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_087
INDEX IN ts_newordA_087
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 74563 ENDING AT 75429,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA88;
CREATE TABLE NEW_ORDERA88
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_088
INDEX IN ts_newordA_088
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 75430 ENDING AT 76296,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA89;

```

```

CREATE TABLE NEW_ORDERA89
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_089
INDEX IN ts_newordA_089
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 76297 ENDING AT 77163,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA90;
CREATE TABLE NEW_ORDERA90
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_090
INDEX IN ts_newordA_090
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 77164 ENDING AT 78030,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA91;
CREATE TABLE NEW_ORDERA91
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_091
INDEX IN ts_newordA_091
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 78031 ENDING AT 78897,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA92;
CREATE TABLE NEW_ORDERA92
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_092
INDEX IN ts_newordA_092
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 78898 ENDING AT 79764,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA93;
CREATE TABLE NEW_ORDERA93
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)

```

```

)
IN ts_newordA_093
INDEX IN ts_newordA_093
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 79765 ENDING AT 80631,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA94;
CREATE TABLE NEW_ORDERA94
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_094
INDEX IN ts_newordA_094
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 80632 ENDING AT 81498,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA95;
CREATE TABLE NEW_ORDERA95
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_095
INDEX IN ts_newordA_095
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 81499 ENDING AT 82365,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA96;
CREATE TABLE NEW_ORDERA96
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_096
INDEX IN ts_newordA_096
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 82366 ENDING AT 83232,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA97;
CREATE TABLE NEW_ORDERA97
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_097
INDEX IN ts_newordA_097
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 83233 ENDING AT 84099,

```

```

NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA98;
CREATE TABLE NEW_ORDERA98
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_098
INDEX IN ts_newordA_098
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 84100 ENDING AT 84966,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA99;
CREATE TABLE NEW_ORDERA99
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_099
INDEX IN ts_newordA_099
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 84967 ENDING AT 85833,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA100;
CREATE TABLE NEW_ORDERA100
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_100
INDEX IN ts_newordA_100
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 85834 ENDING AT 86700,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA101;
CREATE TABLE NEW_ORDERA101
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_101
INDEX IN ts_newordA_101
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 86701 ENDING AT 87567,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERA102;
CREATE TABLE NEW_ORDERA102
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_102
INDEX IN ts_newordA_102
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 87568 ENDING AT 88434,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA103;
CREATE TABLE NEW_ORDERA103
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_103
INDEX IN ts_newordA_103
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 88435 ENDING AT 89301,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA104;
CREATE TABLE NEW_ORDERA104
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_104
INDEX IN ts_newordA_104
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 89302 ENDING AT 90168,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA105;
CREATE TABLE NEW_ORDERA105
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordA_105
INDEX IN ts_newordA_105
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 90169 ENDING AT 91035,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA106;
CREATE TABLE NEW_ORDERA106
(
NO_O_ID INTEGER NOT NULL,

```

```

NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_106
INDEX IN ts_newordA_106
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 91036 ENDING AT 91902,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA107;
CREATE TABLE NEW_ORDERA107
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_107
INDEX IN ts_newordA_107
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 91903 ENDING AT 92769,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA108;
CREATE TABLE NEW_ORDERA108
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_108
INDEX IN ts_newordA_108
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 92770 ENDING AT 93636,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA109;
CREATE TABLE NEW_ORDERA109
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_109
INDEX IN ts_newordA_109
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 93637 ENDING AT 94503,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA110;
CREATE TABLE NEW_ORDERA110
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_110
INDEX IN ts_newordA_110

```

```

ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 94504 ENDING AT 95370,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA111;
CREATE TABLE NEW_ORDERA111
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_111
INDEX IN ts_newordA_111
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 95371 ENDING AT 96237,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA112;
CREATE TABLE NEW_ORDERA112
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_112
INDEX IN ts_newordA_112
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 96238 ENDING AT 97104,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA113;
CREATE TABLE NEW_ORDERA113
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_113
INDEX IN ts_newordA_113
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 97105 ENDING AT 97971,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA114;
CREATE TABLE NEW_ORDERA114
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_114
INDEX IN ts_newordA_114
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 97972 ENDING AT 98838,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA115;
CREATE TABLE NEW_ORDERA115
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_115
INDEX IN ts_newordA_115
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 98839 ENDING AT 99705,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA116;
CREATE TABLE NEW_ORDERA116
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_116
INDEX IN ts_newordA_116
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 99706 ENDING AT 100572,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA117;
CREATE TABLE NEW_ORDERA117
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_117
INDEX IN ts_newordA_117
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 100573 ENDING AT 101439,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA118;
CREATE TABLE NEW_ORDERA118
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordA_118
INDEX IN ts_newordA_118
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 101440 ENDING AT 102306,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA119;
CREATE TABLE NEW_ORDERA119

```

```

(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordA_119
INDEX IN ts_newordA_119
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 102307 ENDING AT 103173,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA120;
CREATE TABLE NEW_ORDERA120
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordA_120
INDEX IN ts_newordA_120
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 103174 ENDING AT 104040,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT 3741
)
)
ALLOW OVERFLOW;
connect reset;

```

### DDL/CRTB NEW ORDERB.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB1;
CREATE TABLE NEW_ORDERB1
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_001
INDEX IN ts_newordB_001
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 1 ENDING AT 867,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB2;
CREATE TABLE NEW_ORDERB2
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_002
INDEX IN ts_newordB_002
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 868 ENDING AT 1734,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB3;
CREATE TABLE NEW_ORDERB3

```

```

(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_003
INDEX IN ts_newordB_003
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 1735 ENDING AT 2601,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB4;
CREATE TABLE NEW_ORDERB4
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_004
INDEX IN ts_newordB_004
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 2602 ENDING AT 3468,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB5;
CREATE TABLE NEW_ORDERB5
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_005
INDEX IN ts_newordB_005
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 3469 ENDING AT 4335,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB6;
CREATE TABLE NEW_ORDERB6
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_006
INDEX IN ts_newordB_006
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 4336 ENDING AT 5202,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB7;
CREATE TABLE NEW_ORDERB7
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)

```

```

IN ts_newordB_007
INDEX IN ts_newordB_007
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 5203 ENDING AT 6069,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB8;
CREATE TABLE NEW_ORDERB8
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_008
INDEX IN ts_newordB_008
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 6070 ENDING AT 6936,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB9;
CREATE TABLE NEW_ORDERB9
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_009
INDEX IN ts_newordB_009
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 6937 ENDING AT 7803,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB10;
CREATE TABLE NEW_ORDERB10
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_010
INDEX IN ts_newordB_010
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 7804 ENDING AT 8670,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB11;
CREATE TABLE NEW_ORDERB11
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT  NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_newordB_011
INDEX IN ts_newordB_011
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 8671 ENDING AT 9537,
NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

        NO_O_ID STARTING FROM 3742 ENDING AT 5583
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB12;
CREATE TABLE NEW_ORDERB12
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_012
INDEX IN ts_newordB_012
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 9538 ENDING AT 10404,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB13;
CREATE TABLE NEW_ORDERB13
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_013
INDEX IN ts_newordB_013
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 10405 ENDING AT 11271,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB14;
CREATE TABLE NEW_ORDERB14
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_014
INDEX IN ts_newordB_014
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 11272 ENDING AT 12138,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB15;
CREATE TABLE NEW_ORDERB15
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_015
INDEX IN ts_newordB_015
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 12139 ENDING AT 13005,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE NEW_ORDERB16;
CREATE TABLE NEW_ORDERB16
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_016
INDEX IN ts_newordB_016
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 13006 ENDING AT 13872,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB17;
CREATE TABLE NEW_ORDERB17
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_017
INDEX IN ts_newordB_017
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 13873 ENDING AT 14739,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB18;
CREATE TABLE NEW_ORDERB18
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_018
INDEX IN ts_newordB_018
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 14740 ENDING AT 15606,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB19;
CREATE TABLE NEW_ORDERB19
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_019
INDEX IN ts_newordB_019
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 15607 ENDING AT 16473,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB20;
CREATE TABLE NEW_ORDERB20
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,

```

```

        NO_W_ID    INTEGER    NOT NULL
    )
    IN ts_newordB_020
    INDEX IN ts_newordB_020
    ORGANIZE BY KEY SEQUENCE (
        NO_W_ID STARTING FROM 16474 ENDING AT 17340,
        NO_D_ID STARTING FROM 1 ENDING AT 10,
        NO_O_ID STARTING FROM 3742 ENDING AT 5583
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB21;
CREATE TABLE NEW_ORDERB21
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_021
INDEX IN ts_newordB_021
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 17341 ENDING AT 18207,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB22;
CREATE TABLE NEW_ORDERB22
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_022
INDEX IN ts_newordB_022
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 18208 ENDING AT 19074,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB23;
CREATE TABLE NEW_ORDERB23
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_023
INDEX IN ts_newordB_023
ORGANIZE BY KEY SEQUENCE (
    NO_W_ID STARTING FROM 19075 ENDING AT 19941,
    NO_D_ID STARTING FROM 1 ENDING AT 10,
    NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB24;
CREATE TABLE NEW_ORDERB24
(
    NO_O_ID    INTEGER    NOT NULL,
    NO_D_ID    SMALLINT  NOT NULL,
    NO_W_ID    INTEGER    NOT NULL
)
IN ts_newordB_024
INDEX IN ts_newordB_024
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 19942 ENDING AT 20808,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB25;
CREATE TABLE NEW_ORDERB25
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_025
INDEX IN ts_newordB_025
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 20809 ENDING AT 21675,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB26;
CREATE TABLE NEW_ORDERB26
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_026
INDEX IN ts_newordB_026
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 21676 ENDING AT 22542,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB27;
CREATE TABLE NEW_ORDERB27
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_027
INDEX IN ts_newordB_027
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 22543 ENDING AT 23409,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB28;
CREATE TABLE NEW_ORDERB28
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_028
INDEX IN ts_newordB_028
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 23410 ENDING AT 24276,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB29;
CREATE TABLE NEW_ORDERB29
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_029
INDEX IN ts_newordB_029
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 24277 ENDING AT 25143,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB30;
CREATE TABLE NEW_ORDERB30
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_030
INDEX IN ts_newordB_030
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 25144 ENDING AT 26010,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB31;
CREATE TABLE NEW_ORDERB31
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_031
INDEX IN ts_newordB_031
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 26011 ENDING AT 26877,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB32;
CREATE TABLE NEW_ORDERB32
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_032
INDEX IN ts_newordB_032
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 26878 ENDING AT 27744,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB33;
CREATE TABLE NEW_ORDERB33
(

```

```

NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_033
INDEX IN ts_newordB_033
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 27745 ENDING AT 28611,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB34;
CREATE TABLE NEW_ORDERB34
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_034
INDEX IN ts_newordB_034
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 28612 ENDING AT 29478,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB35;
CREATE TABLE NEW_ORDERB35
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_035
INDEX IN ts_newordB_035
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 29479 ENDING AT 30345,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB36;
CREATE TABLE NEW_ORDERB36
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_036
INDEX IN ts_newordB_036
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 30346 ENDING AT 31212,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB37;
CREATE TABLE NEW_ORDERB37
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_037

```

```

INDEX IN ts_newordB_037
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 31213 ENDING AT 32079,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB38;
CREATE TABLE NEW_ORDERB38
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_038
INDEX IN ts_newordB_038
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 32080 ENDING AT 32946,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB39;
CREATE TABLE NEW_ORDERB39
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_039
INDEX IN ts_newordB_039
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 32947 ENDING AT 33813,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB40;
CREATE TABLE NEW_ORDERB40
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_040
INDEX IN ts_newordB_040
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 33814 ENDING AT 34680,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB41;
CREATE TABLE NEW_ORDERB41
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_041
INDEX IN ts_newordB_041
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 34681 ENDING AT 35547,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB42;
CREATE TABLE NEW_ORDERB42
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_042
INDEX IN ts_newordB_042
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 35548 ENDING AT 36414,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB43;
CREATE TABLE NEW_ORDERB43
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_043
INDEX IN ts_newordB_043
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 36415 ENDING AT 37281,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB44;
CREATE TABLE NEW_ORDERB44
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_044
INDEX IN ts_newordB_044
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 37282 ENDING AT 38148,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB45;
CREATE TABLE NEW_ORDERB45
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_045
INDEX IN ts_newordB_045
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 38149 ENDING AT 39015,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB46;

```

```

CREATE TABLE NEW_ORDERB46
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_046
INDEX IN ts_newordB_046
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 39016 ENDING AT 39882,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB47;
CREATE TABLE NEW_ORDERB47
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_047
INDEX IN ts_newordB_047
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 39883 ENDING AT 40749,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB48;
CREATE TABLE NEW_ORDERB48
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_048
INDEX IN ts_newordB_048
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 40750 ENDING AT 41616,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB49;
CREATE TABLE NEW_ORDERB49
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_049
INDEX IN ts_newordB_049
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 41617 ENDING AT 42483,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB50;
CREATE TABLE NEW_ORDERB50
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)

```

```

)
IN ts_newordB_050
INDEX IN ts_newordB_050
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 42484 ENDING AT 43350,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB51;
CREATE TABLE NEW_ORDERB51
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_051
INDEX IN ts_newordB_051
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 43351 ENDING AT 44217,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB52;
CREATE TABLE NEW_ORDERB52
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_052
INDEX IN ts_newordB_052
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 44218 ENDING AT 45084,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB53;
CREATE TABLE NEW_ORDERB53
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_053
INDEX IN ts_newordB_053
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 45085 ENDING AT 45951,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB54;
CREATE TABLE NEW_ORDERB54
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_054
INDEX IN ts_newordB_054
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 45952 ENDING AT 46818,

```

```

NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB55;
CREATE TABLE NEW_ORDERB55
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_055
INDEX IN ts_newordB_055
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 46819 ENDING AT 47685,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB56;
CREATE TABLE NEW_ORDERB56
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_056
INDEX IN ts_newordB_056
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 47686 ENDING AT 48552,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB57;
CREATE TABLE NEW_ORDERB57
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_057
INDEX IN ts_newordB_057
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 48553 ENDING AT 49419,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB58;
CREATE TABLE NEW_ORDERB58
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_058
INDEX IN ts_newordB_058
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 49420 ENDING AT 50286,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDERB59;
CREATE TABLE NEW_ORDERB59
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_059
INDEX IN ts_newordB_059
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 50287 ENDING AT 51153,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB60;
CREATE TABLE NEW_ORDERB60
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_060
INDEX IN ts_newordB_060
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 51154 ENDING AT 52020,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB61;
CREATE TABLE NEW_ORDERB61
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_061
INDEX IN ts_newordB_061
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 52021 ENDING AT 52887,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB62;
CREATE TABLE NEW_ORDERB62
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_newordB_062
INDEX IN ts_newordB_062
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 52888 ENDING AT 53754,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB63;
CREATE TABLE NEW_ORDERB63
(
NO_O_ID INTEGER NOT NULL,

```



```

NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_063
INDEX IN ts_newordB_063
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 53755 ENDING AT 54621,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB64;
CREATE TABLE NEW_ORDERB64
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_064
INDEX IN ts_newordB_064
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 54622 ENDING AT 55488,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB65;
CREATE TABLE NEW_ORDERB65
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_065
INDEX IN ts_newordB_065
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 55489 ENDING AT 56355,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB66;
CREATE TABLE NEW_ORDERB66
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_066
INDEX IN ts_newordB_066
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 56356 ENDING AT 57222,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB67;
CREATE TABLE NEW_ORDERB67
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_067
INDEX IN ts_newordB_067

```

```

ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 57223 ENDING AT 58089,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB68;
CREATE TABLE NEW_ORDERB68
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_068
INDEX IN ts_newordB_068
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 58090 ENDING AT 58956,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB69;
CREATE TABLE NEW_ORDERB69
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_069
INDEX IN ts_newordB_069
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 58957 ENDING AT 59823,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB70;
CREATE TABLE NEW_ORDERB70
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_070
INDEX IN ts_newordB_070
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 59824 ENDING AT 60690,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB71;
CREATE TABLE NEW_ORDERB71
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_071
INDEX IN ts_newordB_071
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 60691 ENDING AT 61557,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB72;
CREATE TABLE NEW_ORDERB72
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_072
INDEX IN ts_newordB_072
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 61558 ENDING AT 62424,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB73;
CREATE TABLE NEW_ORDERB73
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_073
INDEX IN ts_newordB_073
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 62425 ENDING AT 63291,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB74;
CREATE TABLE NEW_ORDERB74
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_074
INDEX IN ts_newordB_074
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 63292 ENDING AT 64158,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB75;
CREATE TABLE NEW_ORDERB75
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_075
INDEX IN ts_newordB_075
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 64159 ENDING AT 65025,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB76;
CREATE TABLE NEW_ORDERB76

```

```

(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_076
INDEX IN ts_newordB_076
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 65026 ENDING AT 65892,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB77;
CREATE TABLE NEW_ORDERB77
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_077
INDEX IN ts_newordB_077
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 65893 ENDING AT 66759,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB78;
CREATE TABLE NEW_ORDERB78
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_078
INDEX IN ts_newordB_078
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 66760 ENDING AT 67626,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB79;
CREATE TABLE NEW_ORDERB79
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_079
INDEX IN ts_newordB_079
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 67627 ENDING AT 68493,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB80;
CREATE TABLE NEW_ORDERB80
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)

```

```

IN ts_newordB_080
INDEX IN ts_newordB_080
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 68494 ENDING AT 69360,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB81;
CREATE TABLE NEW_ORDERB81
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_081
INDEX IN ts_newordB_081
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 69361 ENDING AT 70227,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB82;
CREATE TABLE NEW_ORDERB82
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_082
INDEX IN ts_newordB_082
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 70228 ENDING AT 71094,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB83;
CREATE TABLE NEW_ORDERB83
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_083
INDEX IN ts_newordB_083
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 71095 ENDING AT 71961,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB84;
CREATE TABLE NEW_ORDERB84
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_084
INDEX IN ts_newordB_084
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 71962 ENDING AT 72828,
NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB85;
CREATE TABLE NEW_ORDERB85
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_085
INDEX IN ts_newordB_085
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 72829 ENDING AT 73695,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB86;
CREATE TABLE NEW_ORDERB86
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_086
INDEX IN ts_newordB_086
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 73696 ENDING AT 74562,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB87;
CREATE TABLE NEW_ORDERB87
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_087
INDEX IN ts_newordB_087
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 74563 ENDING AT 75429,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB88;
CREATE TABLE NEW_ORDERB88
(
NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
)
IN ts_newordB_088
INDEX IN ts_newordB_088
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 75430 ENDING AT 76296,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE NEW_ORDERB89;
CREATE TABLE NEW_ORDERB89
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_089
INDEX IN ts_newordB_089
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 76297 ENDING AT 77163,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB90;
CREATE TABLE NEW_ORDERB90
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_090
INDEX IN ts_newordB_090
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 77164 ENDING AT 78030,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB91;
CREATE TABLE NEW_ORDERB91
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_091
INDEX IN ts_newordB_091
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 78031 ENDING AT 78897,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB92;
CREATE TABLE NEW_ORDERB92
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_092
INDEX IN ts_newordB_092
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 78898 ENDING AT 79764,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB93;
CREATE TABLE NEW_ORDERB93
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,

```

```

  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_093
INDEX IN ts_newordB_093
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 79765 ENDING AT 80631,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB94;
CREATE TABLE NEW_ORDERB94
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_094
INDEX IN ts_newordB_094
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 80632 ENDING AT 81498,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB95;
CREATE TABLE NEW_ORDERB95
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_095
INDEX IN ts_newordB_095
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 81499 ENDING AT 82365,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB96;
CREATE TABLE NEW_ORDERB96
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_096
INDEX IN ts_newordB_096
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 82366 ENDING AT 83232,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB97;
CREATE TABLE NEW_ORDERB97
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_097
INDEX IN ts_newordB_097
ORGANIZE BY KEY SEQUENCE (

```

```

  NO_W_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_O_ID  INTEGER NOT NULL
)
NO_W_ID STARTING FROM 83233 ENDING AT 84099,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB98;
CREATE TABLE NEW_ORDERB98
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_098
INDEX IN ts_newordB_098
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 84100 ENDING AT 84966,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB99;
CREATE TABLE NEW_ORDERB99
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_099
INDEX IN ts_newordB_099
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 84967 ENDING AT 85833,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB100;
CREATE TABLE NEW_ORDERB100
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_100
INDEX IN ts_newordB_100
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 85834 ENDING AT 86700,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB101;
CREATE TABLE NEW_ORDERB101
(
  NO_O_ID  INTEGER NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER NOT NULL
)
IN ts_newordB_101
INDEX IN ts_newordB_101
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 86701 ENDING AT 87567,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB102;
CREATE TABLE NEW_ORDERB102
(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_102
INDEX IN ts_newordB_102
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 87568 ENDING AT 88434,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB103;
CREATE TABLE NEW_ORDERB103

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_103
INDEX IN ts_newordB_103
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 88435 ENDING AT 89301,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB104;
CREATE TABLE NEW_ORDERB104

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_104
INDEX IN ts_newordB_104
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 89302 ENDING AT 90168,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB105;
CREATE TABLE NEW_ORDERB105

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_105
INDEX IN ts_newordB_105
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 90169 ENDING AT 91035,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB106;
CREATE TABLE NEW_ORDERB106

```

```

(

```

```

NO_O_ID  INTEGER  NOT NULL,
NO_D_ID  SMALLINT NOT NULL,
NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_106
INDEX IN ts_newordB_106
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 91036 ENDING AT 91902,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB107;
CREATE TABLE NEW_ORDERB107

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_107
INDEX IN ts_newordB_107
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 91903 ENDING AT 92769,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB108;
CREATE TABLE NEW_ORDERB108

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_108
INDEX IN ts_newordB_108
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 92770 ENDING AT 93636,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB109;
CREATE TABLE NEW_ORDERB109

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_109
INDEX IN ts_newordB_109
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 93637 ENDING AT 94503,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB110;
CREATE TABLE NEW_ORDERB110

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_110

```

```

INDEX IN ts_newordB_110
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 94504 ENDING AT 95370,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB111;
CREATE TABLE NEW_ORDERB111

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_111
INDEX IN ts_newordB_111
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 95371 ENDING AT 96237,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB112;
CREATE TABLE NEW_ORDERB112

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_112
INDEX IN ts_newordB_112
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 96238 ENDING AT 97104,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB113;
CREATE TABLE NEW_ORDERB113

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_113
INDEX IN ts_newordB_113
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 97105 ENDING AT 97971,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB114;
CREATE TABLE NEW_ORDERB114

```

```

(
  NO_O_ID  INTEGER  NOT NULL,
  NO_D_ID  SMALLINT NOT NULL,
  NO_W_ID  INTEGER  NOT NULL
)
IN ts_newordB_114
INDEX IN ts_newordB_114
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 97972 ENDING AT 98838,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3742 ENDING AT 5583
)

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB115;
CREATE TABLE NEW_ORDERB115
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_115
INDEX IN ts_newordB_115
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 98839 ENDING AT 99705,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB116;
CREATE TABLE NEW_ORDERB116
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_116
INDEX IN ts_newordB_116
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 99706 ENDING AT 100572,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB117;
CREATE TABLE NEW_ORDERB117
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_117
INDEX IN ts_newordB_117
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 100573 ENDING AT 101439,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB118;
CREATE TABLE NEW_ORDERB118
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_118
INDEX IN ts_newordB_118
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 101440 ENDING AT 102306,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB119;

```

```

CREATE TABLE NEW_ORDERB119
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_119
INDEX IN ts_newordB_119
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 102307 ENDING AT 103173,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB120;
CREATE TABLE NEW_ORDERB120
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
IN ts_newordB_120
INDEX IN ts_newordB_120
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 103174 ENDING AT 104040,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3742 ENDING AT 5583
)
ALLOW OVERFLOW;
connect reset;

```

### DDL/CRTB\_ORDERS.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS1;
CREATE TABLE ORDERS1
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_001
INDEX IN is_order_001
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 1 ENDING AT 867,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS2;
CREATE TABLE ORDERS2
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_002

```

```

INDEX IN is_order_002
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 868 ENDING AT 1734,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS3;
CREATE TABLE ORDERS3
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_003
INDEX IN is_order_003
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 1735 ENDING AT 2601,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS4;
CREATE TABLE ORDERS4
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_004
INDEX IN is_order_004
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 2602 ENDING AT 3468,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS5;
CREATE TABLE ORDERS5
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_005
INDEX IN is_order_005
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 3469 ENDING AT 4335,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS6;
CREATE TABLE ORDERS6
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_006
INDEX IN is_order_006
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 4336 ENDING AT 5202,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS7;
CREATE TABLE ORDERS7
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_007
INDEX IN is_order_007
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 5203 ENDING AT 6069,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS8;
CREATE TABLE ORDERS8
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_008
INDEX IN is_order_008
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 6070 ENDING AT 6936,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS9;
CREATE TABLE ORDERS9
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,

```

```

O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_009
INDEX IN is_order_009
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 6937 ENDING AT 7803,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS10;
CREATE TABLE ORDERS10
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_010
INDEX IN is_order_010
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 7804 ENDING AT 8670,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS11;
CREATE TABLE ORDERS11
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_011
INDEX IN is_order_011
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 8671 ENDING AT 9537,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS12;
CREATE TABLE ORDERS12
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)

```

```

IN ts_order_012
INDEX IN is_order_012
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 9538 ENDING AT 10404,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS13;
CREATE TABLE ORDERS13
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_013
INDEX IN is_order_013
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 10405 ENDING AT 11271,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS14;
CREATE TABLE ORDERS14
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_014
INDEX IN is_order_014
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 11272 ENDING AT 12138,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS15;
CREATE TABLE ORDERS15
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
IN ts_order_015
INDEX IN is_order_015
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 12139 ENDING AT 13005,
  O_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS16;
CREATE TABLE ORDERS16
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_016
INDEX IN is_order_016
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 13006 ENDING AT 13872,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS17;
CREATE TABLE ORDERS17
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_017
INDEX IN is_order_017
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 13873 ENDING AT 14739,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS18;
CREATE TABLE ORDERS18
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_018
INDEX IN is_order_018
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 14740 ENDING AT 15606,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS19;
CREATE TABLE ORDERS19
(
O_C_ID INTEGER NOT NULL,

```

```

O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_019
INDEX IN is_order_019
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 15607 ENDING AT 16473,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS20;
CREATE TABLE ORDERS20
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_020
INDEX IN is_order_020
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 16474 ENDING AT 17340,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS21;
CREATE TABLE ORDERS21
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_021
INDEX IN is_order_021
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 17341 ENDING AT 18207,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS22;
CREATE TABLE ORDERS22
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)

```

```

)
IN ts_order_022
INDEX IN is_order_022
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 18208 ENDING AT 19074,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS23;
CREATE TABLE ORDERS23
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_023
INDEX IN is_order_023
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 19075 ENDING AT 19941,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS24;
CREATE TABLE ORDERS24
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_024
INDEX IN is_order_024
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 19942 ENDING AT 20808,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS25;
CREATE TABLE ORDERS25
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_025
INDEX IN is_order_025
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 20809 ENDING AT 21675,
O_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS26;
CREATE TABLE ORDERS26
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_026
INDEX IN is_order_026
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 21676 ENDING AT 22542,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS27;
CREATE TABLE ORDERS27
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_027
INDEX IN is_order_027
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 22543 ENDING AT 23409,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS28;
CREATE TABLE ORDERS28
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_028
INDEX IN is_order_028
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 23410 ENDING AT 24276,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS29;
CREATE TABLE ORDERS29
(

```

```

O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_029
INDEX IN is_order_029
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 24277 ENDING AT 25143,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS30;
CREATE TABLE ORDERS30
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_030
INDEX IN is_order_030
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 25144 ENDING AT 26010,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS31;
CREATE TABLE ORDERS31
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_031
INDEX IN is_order_031
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 26011 ENDING AT 26877,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS32;
CREATE TABLE ORDERS32
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,

```

```

O_D_ID SMALLINT NOT NULL
)
IN ts_order_032
INDEX IN is_order_032
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 26878 ENDING AT 27744,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS33;
CREATE TABLE ORDERS33
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_033
INDEX IN is_order_033
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 27745 ENDING AT 28611,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS34;
CREATE TABLE ORDERS34
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_034
INDEX IN is_order_034
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 28612 ENDING AT 29478,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS35;
CREATE TABLE ORDERS35
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_035
INDEX IN is_order_035
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 29479 ENDING AT 30345,

```



```

        O_D_ID STARTING FROM 1 ENDING AT 10
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS36;
CREATE TABLE ORDERS36
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_036
INDEX IN is_order_036
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 30346 ENDING AT 31212,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS37;
CREATE TABLE ORDERS37
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_037
INDEX IN is_order_037
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 31213 ENDING AT 32079,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS38;
CREATE TABLE ORDERS38
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_038
INDEX IN is_order_038
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 32080 ENDING AT 32946,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS39;
CREATE TABLE ORDERS39

```

```

(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_039
INDEX IN is_order_039
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 32947 ENDING AT 33813,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS40;
CREATE TABLE ORDERS40
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_040
INDEX IN is_order_040
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 33814 ENDING AT 34680,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS41;
CREATE TABLE ORDERS41
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_041
INDEX IN is_order_041
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 34681 ENDING AT 35547,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS42;
CREATE TABLE ORDERS42
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,

```

```

    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_042
INDEX IN is_order_042
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 35548 ENDING AT 36414,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS43;
CREATE TABLE ORDERS43
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_043
INDEX IN is_order_043
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 36415 ENDING AT 37281,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS44;
CREATE TABLE ORDERS44
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_044
INDEX IN is_order_044
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,
    O_W_ID STARTING FROM 37282 ENDING AT 38148,
    O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS45;
CREATE TABLE ORDERS45
(
    O_C_ID INTEGER NOT NULL,
    O_ENTRY_D TIMESTAMP NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT SMALLINT NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID INTEGER NOT NULL,
    O_W_ID INTEGER NOT NULL,
    O_D_ID SMALLINT NOT NULL
)
IN ts_order_045
INDEX IN is_order_045
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 1 ENDING AT 3741,

```

```

O_W_ID STARTING FROM 38149 ENDING AT 39015,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS46;
CREATE TABLE ORDERS46
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_046
INDEX IN is_order_046
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 39016 ENDING AT 39882,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS47;
CREATE TABLE ORDERS47
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_047
INDEX IN is_order_047
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 39883 ENDING AT 40749,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS48;
CREATE TABLE ORDERS48
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_048
INDEX IN is_order_048
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 40750 ENDING AT 41616,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS49;

```

```

CREATE TABLE ORDERS49
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_049
INDEX IN is_order_049
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 41617 ENDING AT 42483,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS50;
CREATE TABLE ORDERS50
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_050
INDEX IN is_order_050
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 42484 ENDING AT 43350,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS51;
CREATE TABLE ORDERS51
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_051
INDEX IN is_order_051
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 43351 ENDING AT 44217,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS52;
CREATE TABLE ORDERS52
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,

```

```

O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_052
INDEX IN is_order_052
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 44218 ENDING AT 45084,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS53;
CREATE TABLE ORDERS53
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_053
INDEX IN is_order_053
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 45085 ENDING AT 45951,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS54;
CREATE TABLE ORDERS54
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_054
INDEX IN is_order_054
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 45952 ENDING AT 46818,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS55;
CREATE TABLE ORDERS55
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_055
INDEX IN is_order_055
ORGANIZE BY KEY SEQUENCE (

```

```

O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 46819 ENDING AT 47685,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS56;
CREATE TABLE ORDERS56
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_056
INDEX IN is_order_056
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 47686 ENDING AT 48552,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS57;
CREATE TABLE ORDERS57
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_057
INDEX IN is_order_057
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 48553 ENDING AT 49419,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS58;
CREATE TABLE ORDERS58
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_058
INDEX IN is_order_058
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 49420 ENDING AT 50286,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE ORDERS59;
CREATE TABLE ORDERS59
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_059
INDEX IN is_order_059
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 50287 ENDING AT 51153,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS60;
CREATE TABLE ORDERS60
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_060
INDEX IN is_order_060
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 51154 ENDING AT 52020,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS61;
CREATE TABLE ORDERS61
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_061
INDEX IN is_order_061
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 52021 ENDING AT 52887,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS62;
CREATE TABLE ORDERS62
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,

```

```

O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_062
INDEX IN is_order_062
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 52888 ENDING AT 53754,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS63;
CREATE TABLE ORDERS63
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_063
INDEX IN is_order_063
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 53755 ENDING AT 54621,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS64;
CREATE TABLE ORDERS64
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_064
INDEX IN is_order_064
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 54622 ENDING AT 55488,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS65;
CREATE TABLE ORDERS65
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_065
INDEX IN is_order_065

```

```

ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 55489 ENDING AT 56355,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS66;
CREATE TABLE ORDERS66
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_066
INDEX IN is_order_066
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 56356 ENDING AT 57222,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS67;
CREATE TABLE ORDERS67
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_067
INDEX IN is_order_067
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 57223 ENDING AT 58089,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS68;
CREATE TABLE ORDERS68
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_068
INDEX IN is_order_068
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 58090 ENDING AT 58956,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE ORDERS69;
CREATE TABLE ORDERS69
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_069
INDEX IN is_order_069
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 58957 ENDING AT 59823,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS70;
CREATE TABLE ORDERS70
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_070
INDEX IN is_order_070
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 59824 ENDING AT 60690,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS71;
CREATE TABLE ORDERS71
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_071
INDEX IN is_order_071
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 60691 ENDING AT 61557,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS72;
CREATE TABLE ORDERS72
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,

```

```

O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_072
INDEX IN is_order_072
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 61558 ENDING AT 62424,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS73;
CREATE TABLE ORDERS73
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_073
INDEX IN is_order_073
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 62425 ENDING AT 63291,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS74;
CREATE TABLE ORDERS74
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_074
INDEX IN is_order_074
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 63292 ENDING AT 64158,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS75;
CREATE TABLE ORDERS75
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D TIMESTAMP NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_075

```

```

INDEX IN is_order_075
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 64159 ENDING AT 65025,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS76;
CREATE TABLE ORDERS76
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
)
IN ts_order_076
INDEX IN is_order_076
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 65026 ENDING AT 65892,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS77;
CREATE TABLE ORDERS77
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
)
IN ts_order_077
INDEX IN is_order_077
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 65893 ENDING AT 66759,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS78;
CREATE TABLE ORDERS78
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
)
IN ts_order_078
INDEX IN is_order_078
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 66760 ENDING AT 67626,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS79;
CREATE TABLE ORDERS79
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
)
IN ts_order_079
INDEX IN is_order_079
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 67627 ENDING AT 68493,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS80;
CREATE TABLE ORDERS80
(
  O_C_ID INTEGER NOT NULL,
  O_ENTRY_D TIMESTAMP NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID INTEGER NOT NULL,
  O_W_ID INTEGER NOT NULL,
  O_D_ID SMALLINT NOT NULL
)
)
IN ts_order_080
INDEX IN is_order_080
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 68494 ENDING AT 69360,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS81;
CREATE TABLE ORDERS81
(
  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_081
INDEX IN is_order_081
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 69361 ENDING AT 70227,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS82;
CREATE TABLE ORDERS82
(
  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_082
INDEX IN is_order_082
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 70228 ENDING AT 71094,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS83;
CREATE TABLE ORDERS83
(
  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_083
INDEX IN is_order_083
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 71095 ENDING AT 71961,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS84;
CREATE TABLE ORDERS84
(
  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_084
INDEX IN is_order_084
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 71962 ENDING AT 72828,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS85;
CREATE TABLE ORDERS85
(
  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_085
INDEX IN is_order_085
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 72829 ENDING AT 73695,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS86;
CREATE TABLE ORDERS86
(
  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_086
INDEX IN is_order_086
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 73696 ENDING AT 74562,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS87;
CREATE TABLE ORDERS87
(
  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_087
INDEX IN is_order_087
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 74563 ENDING AT 75429,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS88;
CREATE TABLE ORDERS88
(
  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_088
INDEX IN is_order_088
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 75430 ENDING AT 76296,
  O_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS89;
CREATE TABLE ORDERS89
(
  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 is_order_089
INDEX IN is_order_089
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 76297 ENDING AT 77163,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS90;
CREATE TABLE ORDERS90
(
  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_090
INDEX IN is_order_090
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 77164 ENDING AT 78030,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS91;
CREATE TABLE ORDERS91
(
  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_091
INDEX IN is_order_091
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 78031 ENDING AT 78897,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS92;
CREATE TABLE ORDERS92
(
  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_092
INDEX IN is_order_092
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 78898 ENDING AT 79764,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS93;
CREATE TABLE ORDERS93
(
  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_093
INDEX IN is_order_093
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 79765 ENDING AT 80631,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS94;
CREATE TABLE ORDERS94
(
  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_094
INDEX IN is_order_094
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 80632 ENDING AT 81498,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS95;
CREATE TABLE ORDERS95
(
  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_095
INDEX IN is_order_095
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 81499 ENDING AT 82365,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS96;
CREATE TABLE ORDERS96
(
  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_096
INDEX IN is_order_096
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 82366 ENDING AT 83232,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS97;
CREATE TABLE ORDERS97
(
  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_097
INDEX IN is_order_097
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 83233 ENDING AT 84099,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS98;
CREATE TABLE ORDERS98
(
  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_098
INDEX IN is_order_098
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 84100 ENDING AT 84966,
  O_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS99;
CREATE TABLE ORDERS99
(
  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_099
INDEX IN is_order_099
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 84967 ENDING AT 85833,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS100;
CREATE TABLE ORDERS100
(
  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_100
INDEX IN is_order_100
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 85834 ENDING AT 86700,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS101;
CREATE TABLE ORDERS101
(
  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_101
INDEX IN is_order_101
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 86701 ENDING AT 87567,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS102;
CREATE TABLE ORDERS102
(

```

```

  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_102
INDEX IN is_order_102
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 87568 ENDING AT 88434,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS103;
CREATE TABLE ORDERS103
(
  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_103
INDEX IN is_order_103
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 88435 ENDING AT 89301,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS104;
CREATE TABLE ORDERS104
(
  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_104
INDEX IN is_order_104
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 1 ENDING AT 3741,
  O_W_ID STARTING FROM 89302 ENDING AT 90168,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS105;
CREATE TABLE ORDERS105
(
  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_105
INDEX IN is_order_105
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 90169 ENDING AT 91035,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS106;
CREATE TABLE ORDERS106
(
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_106
INDEX IN is_order_106
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 91036 ENDING AT 91902,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS107;
CREATE TABLE ORDERS107
(
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_107
INDEX IN is_order_107
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 91903 ENDING AT 92769,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS108;
CREATE TABLE ORDERS108
(
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_108
INDEX IN is_order_108
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 92770 ENDING AT 93636,

```

```

O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS109;
CREATE TABLE ORDERS109
(
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_109
INDEX IN is_order_109
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 93637 ENDING AT 94503,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS110;
CREATE TABLE ORDERS110
(
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_110
INDEX IN is_order_110
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 94504 ENDING AT 95370,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS111;
CREATE TABLE ORDERS111
(
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_111
INDEX IN is_order_111
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 95371 ENDING AT 96237,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS112;
CREATE TABLE ORDERS112

```

```

(
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_112
INDEX IN is_order_112
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 96238 ENDING AT 97104,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS113;
CREATE TABLE ORDERS113
(
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_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_113
INDEX IN is_order_113
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 97105 ENDING AT 97971,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS114;
CREATE TABLE ORDERS114
(
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_114
INDEX IN is_order_114
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 97972 ENDING AT 98838,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS115;
CREATE TABLE ORDERS115
(
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_115
INDEX IN is_order_115
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 98839 ENDING AT 99705,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS116;
CREATE TABLE ORDERS116
(
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_116
INDEX IN is_order_116
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 99706 ENDING AT 100572,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS117;
CREATE TABLE ORDERS117
(
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_117
INDEX IN is_order_117
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 100573 ENDING AT 101439,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS118;
CREATE TABLE ORDERS118
(
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_118
INDEX IN is_order_118
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,

```

```

O_W_ID STARTING FROM 101440 ENDING AT 102306,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS119;
CREATE TABLE ORDERS119
(
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_119
INDEX IN is_order_119
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 102307 ENDING AT 103173,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS120;
CREATE TABLE ORDERS120
(
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_120
INDEX IN is_order_120
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3741,
O_W_ID STARTING FROM 103174 ENDING AT 104040,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

DDL/CRTB ORDER LINE.ddl

connect to TPCC in share mode;
DROP TABLE ORDER_LINE1;
CREATE TABLE ORDER_LINE1
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_001
INDEX IN ts_orderline_001
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 1 ENDING AT 867,

```

```

OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE2;
CREATE TABLE ORDER_LINE2
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_002
INDEX IN ts_orderline_002
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 868 ENDING AT 1734,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE3;
CREATE TABLE ORDER_LINE3
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_003
INDEX IN ts_orderline_003
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 1735 ENDING AT 2601,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE4;
CREATE TABLE ORDER_LINE4
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_004
INDEX IN ts_orderline_004

```

```

ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 2602 ENDING AT 3468,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE5;
CREATE TABLE ORDER_LINE5
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_005
INDEX IN ts_orderline_005
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 3469 ENDING AT 4335,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE6;
CREATE TABLE ORDER_LINE6
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_006
INDEX IN ts_orderline_006
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 4336 ENDING AT 5202,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE7;
CREATE TABLE ORDER_LINE7
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)

```

```

IN ts_orderline_007
INDEX IN ts_orderline_007
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 5203 ENDING AT 6069,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE8;
CREATE TABLE ORDER_LINE8
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_008
INDEX IN ts_orderline_008
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 6070 ENDING AT 6936,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE9;
CREATE TABLE ORDER_LINE9
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_009
INDEX IN ts_orderline_009
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 6937 ENDING AT 7803,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE10;
CREATE TABLE ORDER_LINE10
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL,

```

```

OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_010
INDEX IN ts_orderline_010
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 7804 ENDING AT 8670,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE11;
CREATE TABLE ORDER_LINE11
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_011
INDEX IN ts_orderline_011
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 8671 ENDING AT 9537,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE12;
CREATE TABLE ORDER_LINE12
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_012
INDEX IN ts_orderline_012
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 9538 ENDING AT 10404,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE13;
CREATE TABLE ORDER_LINE13
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,

```

```

OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_013
INDEX IN ts_orderline_013
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 10405 ENDING AT 11271,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE14;
CREATE TABLE ORDER_LINE14
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_014
INDEX IN ts_orderline_014
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 11272 ENDING AT 12138,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE15;
CREATE TABLE ORDER_LINE15
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_015
INDEX IN ts_orderline_015
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 12139 ENDING AT 13005,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE16;
CREATE TABLE ORDER_LINE16
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,

```

```

OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_016
INDEX IN ts_orderline_016
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 13006 ENDING AT 13872,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE17;
CREATE TABLE ORDER_LINE17
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_017
INDEX IN ts_orderline_017
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 13873 ENDING AT 14739,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE18;
CREATE TABLE ORDER_LINE18
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_018
INDEX IN ts_orderline_018
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 14740 ENDING AT 15606,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE19;
CREATE TABLE ORDER_LINE19
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,

```

```

OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_019
INDEX IN ts_orderline_019
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 15607 ENDING AT 16473,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE20;
CREATE TABLE ORDER_LINE20
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_020
INDEX IN ts_orderline_020
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 16474 ENDING AT 17340,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE21;
CREATE TABLE ORDER_LINE21
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_021
INDEX IN ts_orderline_021
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 17341 ENDING AT 18207,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE22;
CREATE TABLE ORDER_LINE22
(
OL_DELIVERY_D TIMESTAMP NOT NULL,

```

```

OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_022
INDEX IN ts_orderline_022
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 18208 ENDING AT 19074,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE23;
CREATE TABLE ORDER_LINE23
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_023
INDEX IN ts_orderline_023
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 19075 ENDING AT 19941,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE24;
CREATE TABLE ORDER_LINE24
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_024
INDEX IN ts_orderline_024
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 19942 ENDING AT 20808,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE25;
CREATE TABLE ORDER_LINE25

```

```

(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_025
INDEX IN ts_orderline_025
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 20809 ENDING AT 21675,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE26;
CREATE TABLE ORDER_LINE26
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_026
INDEX IN ts_orderline_026
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 21676 ENDING AT 22542,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE27;
CREATE TABLE ORDER_LINE27
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_027
INDEX IN ts_orderline_027
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 22543 ENDING AT 23409,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE ORDER_LINE28;
CREATE TABLE ORDER_LINE28
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_028
INDEX IN ts_orderline_028
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 23410 ENDING AT 24276,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE29;
CREATE TABLE ORDER_LINE29
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_029
INDEX IN ts_orderline_029
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 24277 ENDING AT 25143,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE30;
CREATE TABLE ORDER_LINE30
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_030
INDEX IN ts_orderline_030
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 25144 ENDING AT 26010,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE31;
CREATE TABLE ORDER_LINE31
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_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 26011 ENDING AT 26877,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE32;
CREATE TABLE ORDER_LINE32
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_032
INDEX IN ts_orderline_032
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 26878 ENDING AT 27744,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE33;
CREATE TABLE ORDER_LINE33
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_033
INDEX IN ts_orderline_033
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 27745 ENDING AT 28611,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE34;
CREATE TABLE ORDER_LINE34
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_034
INDEX IN ts_orderline_034
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 28612 ENDING AT 29478,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE35;
CREATE TABLE ORDER_LINE35
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_035
INDEX IN ts_orderline_035
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 29479 ENDING AT 30345,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE36;
CREATE TABLE ORDER_LINE36
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_036
INDEX IN ts_orderline_036
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 30346 ENDING AT 31212,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
)

```

```

  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE37;
CREATE TABLE ORDER_LINE37
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_037
INDEX IN ts_orderline_037
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 31213 ENDING AT 32079,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE38;
CREATE TABLE ORDER_LINE38
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_038
INDEX IN ts_orderline_038
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 32080 ENDING AT 32946,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE39;
CREATE TABLE ORDER_LINE39
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_039
INDEX IN ts_orderline_039
ORGANIZE BY KEY SEQUENCE (
)

```

```

OL_W_ID STARTING FROM 32947 ENDING AT 33813,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE40;
CREATE TABLE ORDER_LINE40
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_040
INDEX IN ts_orderline_040
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 33814 ENDING AT 34680,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE41;
CREATE TABLE ORDER_LINE41
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_041
INDEX IN ts_orderline_041
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 34681 ENDING AT 35547,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE42;
CREATE TABLE ORDER_LINE42
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_042

```

```

INDEX IN ts_orderline_042
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 35548 ENDING AT 36414,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE43;
CREATE TABLE ORDER_LINE43
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_043
INDEX IN ts_orderline_043
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 36415 ENDING AT 37281,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE44;
CREATE TABLE ORDER_LINE44
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_044
INDEX IN ts_orderline_044
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 37282 ENDING AT 38148,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE45;
CREATE TABLE ORDER_LINE45
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)

```

```

)
IN ts_orderline_045
INDEX IN ts_orderline_045
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 38149 ENDING AT 39015,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE46;
CREATE TABLE ORDER_LINE46
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_046
INDEX IN ts_orderline_046
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 39016 ENDING AT 39882,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE47;
CREATE TABLE ORDER_LINE47
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_047
INDEX IN ts_orderline_047
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 39883 ENDING AT 40749,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE48;
CREATE TABLE ORDER_LINE48
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,

```

```

OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_048
INDEX IN ts_orderline_048
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 40750 ENDING AT 41616,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE49;
CREATE TABLE ORDER_LINE49
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_049
INDEX IN ts_orderline_049
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 41617 ENDING AT 42483,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE50;
CREATE TABLE ORDER_LINE50
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_050
INDEX IN ts_orderline_050
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 42484 ENDING AT 43350,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE51;
CREATE TABLE ORDER_LINE51
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,

```

```

OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
)
IN ts_orderline_051
INDEX IN ts_orderline_051
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 43351 ENDING AT 44217,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE52;
CREATE TABLE ORDER_LINE52
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_052
INDEX IN ts_orderline_052
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 44218 ENDING AT 45084,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE53;
CREATE TABLE ORDER_LINE53
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_053
INDEX IN ts_orderline_053
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 45085 ENDING AT 45951,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE54;
CREATE TABLE ORDER_LINE54
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
)
IN ts_orderline_054
INDEX IN ts_orderline_054
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 45952 ENDING AT 46818,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE55;
CREATE TABLE ORDER_LINE55
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_055
INDEX IN ts_orderline_055
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 46819 ENDING AT 47685,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE56;
CREATE TABLE ORDER_LINE56
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_056
INDEX IN ts_orderline_056
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 47686 ENDING AT 48552,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE57;
CREATE TABLE ORDER_LINE57
(
OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,

```

```

OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_057
INDEX IN ts_orderline_057
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 48553 ENDING AT 49419,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE58;
CREATE TABLE ORDER_LINE58

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_058
INDEX IN ts_orderline_058
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 49420 ENDING AT 50286,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE59;
CREATE TABLE ORDER_LINE59

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_059
INDEX IN ts_orderline_059
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 50287 ENDING AT 51153,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE60;
CREATE TABLE ORDER_LINE60

```

```
(
```

```

OL_DELIVERY_D TIMESTAMP NOT NULL,
OL_AMOUNT DECIMAL(6,2) NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_060
INDEX IN ts_orderline_060
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 51154 ENDING AT 52020,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE61;
CREATE TABLE ORDER_LINE61

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_061
INDEX IN ts_orderline_061
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 52021 ENDING AT 52887,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE62;
CREATE TABLE ORDER_LINE62

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_062
INDEX IN ts_orderline_062
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 52888 ENDING AT 53754,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE63;

```

```

CREATE TABLE ORDER_LINE63
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_063
INDEX IN ts_orderline_063
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 53755 ENDING AT 54621,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE64;
CREATE TABLE ORDER_LINE64

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_064
INDEX IN ts_orderline_064
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 54622 ENDING AT 55488,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE65;
CREATE TABLE ORDER_LINE65

```

```

(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_065
INDEX IN ts_orderline_065
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 55489 ENDING AT 56355,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

```



```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE66;
CREATE TABLE ORDER_LINE66
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_066
INDEX IN ts_orderline_066
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 56356 ENDING AT 57222,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE67;
CREATE TABLE ORDER_LINE67
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_067
INDEX IN ts_orderline_067
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 57223 ENDING AT 58089,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE68;
CREATE TABLE ORDER_LINE68
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_068
INDEX IN ts_orderline_068
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 58090 ENDING AT 58956,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE69;
CREATE TABLE ORDER_LINE69
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_069
INDEX IN ts_orderline_069
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 58957 ENDING AT 59823,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE70;
CREATE TABLE ORDER_LINE70
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_070
INDEX IN ts_orderline_070
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 59824 ENDING AT 60690,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE71;
CREATE TABLE ORDER_LINE71
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_071
INDEX IN ts_orderline_071
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 60691 ENDING AT 61557,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,

```

```

  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE72;
CREATE TABLE ORDER_LINE72
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_072
INDEX IN ts_orderline_072
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 61558 ENDING AT 62424,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE73;
CREATE TABLE ORDER_LINE73
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_073
INDEX IN ts_orderline_073
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 62425 ENDING AT 63291,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE74;
CREATE TABLE ORDER_LINE74
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_074
INDEX IN ts_orderline_074
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 63292 ENDING AT 64158,

```

```

OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE75;
CREATE TABLE ORDER_LINE75
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_075
INDEX IN ts_orderline_075
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 64159 ENDING AT 65025,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE76;
CREATE TABLE ORDER_LINE76
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_076
INDEX IN ts_orderline_076
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 65026 ENDING AT 65892,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE77;
CREATE TABLE ORDER_LINE77
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_077
INDEX IN ts_orderline_077

```

```

ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 65893 ENDING AT 66759,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE78;
CREATE TABLE ORDER_LINE78
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_078
INDEX IN ts_orderline_078
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 66760 ENDING AT 67626,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE79;
CREATE TABLE ORDER_LINE79
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_079
INDEX IN ts_orderline_079
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 67627 ENDING AT 68493,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE80;
CREATE TABLE ORDER_LINE80
(
  OL_DELIVERY_D TIMESTAMP NOT NULL,
  OL_AMOUNT DECIMAL(6,2) NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)

```

```

IN ts_orderline_080
INDEX IN ts_orderline_080
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 68494 ENDING AT 69360,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE81;
CREATE TABLE ORDER_LINE81
(
  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_081
INDEX IN ts_orderline_081
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 69361 ENDING AT 70227,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE82;
CREATE TABLE ORDER_LINE82
(
  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_082
INDEX IN ts_orderline_082
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 70228 ENDING AT 71094,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE83;
CREATE TABLE ORDER_LINE83
(
  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_083
INDEX IN ts_orderline_083
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 71095 ENDING AT 71961,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE84;
CREATE TABLE ORDER_LINE84
(
  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_084
INDEX IN ts_orderline_084
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 71962 ENDING AT 72828,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE85;
CREATE TABLE ORDER_LINE85
(
  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_085
INDEX IN ts_orderline_085
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 72829 ENDING AT 73695,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE86;
CREATE TABLE ORDER_LINE86
(
  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_086
INDEX IN ts_orderline_086
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 73696 ENDING AT 74562,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE87;
CREATE TABLE ORDER_LINE87
(
  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_087
INDEX IN ts_orderline_087
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 74563 ENDING AT 75429,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE88;
CREATE TABLE ORDER_LINE88
(
  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_088
INDEX IN ts_orderline_088
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 75430 ENDING AT 76296,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE89;
CREATE TABLE ORDER_LINE89
(
  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_089
INDEX IN ts_orderline_089
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 76297 ENDING AT 77163,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE90;
CREATE TABLE ORDER_LINE90
(
  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_090
INDEX IN ts_orderline_090
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 77164 ENDING AT 78030,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE91;
CREATE TABLE ORDER_LINE91
(
  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_091
INDEX IN ts_orderline_091
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 78031 ENDING AT 78897,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE92;
CREATE TABLE ORDER_LINE92
(
  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_092
INDEX IN ts_orderline_092
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 78898 ENDING AT 79764,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE93;
CREATE TABLE ORDER_LINE93
(
  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_093
INDEX IN ts_orderline_093
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 79765 ENDING AT 80631,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE94;
CREATE TABLE ORDER_LINE94
(
  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_094
INDEX IN ts_orderline_094
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 80632 ENDING AT 81498,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE95;
CREATE TABLE ORDER_LINE95
(
  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_095
INDEX IN ts_orderline_095
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 81499 ENDING AT 82365,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE96;
CREATE TABLE ORDER_LINE96
(
  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_096
INDEX IN ts_orderline_096
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 82366 ENDING AT 83232,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE97;
CREATE TABLE ORDER_LINE97
(
  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_097
INDEX IN ts_orderline_097
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 83233 ENDING AT 84099,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE98;
CREATE TABLE ORDER_LINE98

```

```

(
  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_098
INDEX IN ts_orderline_098
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 84100 ENDING AT 84966,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE99;
CREATE TABLE ORDER_LINE99
(
  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_099
INDEX IN ts_orderline_099
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 84967 ENDING AT 85833,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE100;
CREATE TABLE ORDER_LINE100
(
  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_100
INDEX IN ts_orderline_100
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 85834 ENDING AT 86700,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE ORDER_LINE101;
CREATE TABLE ORDER_LINE101
(
  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_101
INDEX IN ts_orderline_101
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 86701 ENDING AT 87567,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE102;
CREATE TABLE ORDER_LINE102
(
  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_102
INDEX IN ts_orderline_102
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 87568 ENDING AT 88434,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE103;
CREATE TABLE ORDER_LINE103
(
  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_103
INDEX IN ts_orderline_103
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 88435 ENDING AT 89301,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE104;
CREATE TABLE ORDER_LINE104
(
  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_104
INDEX IN ts_orderline_104
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 89302 ENDING AT 90168,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE105;
CREATE TABLE ORDER_LINE105
(
  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_105
INDEX IN ts_orderline_105
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 90169 ENDING AT 91035,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE106;
CREATE TABLE ORDER_LINE106
(
  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_106
INDEX IN ts_orderline_106
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 91036 ENDING AT 91902,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE107;
CREATE TABLE ORDER_LINE107
(
  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_107
INDEX IN ts_orderline_107
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 91903 ENDING AT 92769,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE108;
CREATE TABLE ORDER_LINE108
(
  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_108
INDEX IN ts_orderline_108
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 92770 ENDING AT 93636,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE109;
CREATE TABLE ORDER_LINE109
(
  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_109
INDEX IN ts_orderline_109
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 93637 ENDING AT 94503,
  OL_D_ID STARTING FROM 1 ENDING AT 10,

```

```

OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE110;
CREATE TABLE ORDER_LINE110
(
  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_110
INDEX IN ts_orderline_110
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 94504 ENDING AT 95370,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE111;
CREATE TABLE ORDER_LINE111
(
  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_111
INDEX IN ts_orderline_111
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 95371 ENDING AT 96237,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE112;
CREATE TABLE ORDER_LINE112
(
  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_112
INDEX IN ts_orderline_112
ORGANIZE BY KEY SEQUENCE (

```

```

OL_W_ID STARTING FROM 96238 ENDING AT 97104,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3741,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE113;
CREATE TABLE ORDER_LINE113
(
  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_113
INDEX IN ts_orderline_113
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 97105 ENDING AT 97971,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE114;
CREATE TABLE ORDER_LINE114
(
  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_114
INDEX IN ts_orderline_114
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 97972 ENDING AT 98838,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE115;
CREATE TABLE ORDER_LINE115
(
  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_115

```

```

INDEX IN ts_orderline_115
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 98839 ENDING AT 99705,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE116;
CREATE TABLE ORDER_LINE116
(
  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_116
INDEX IN ts_orderline_116
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 99706 ENDING AT 100572,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE117;
CREATE TABLE ORDER_LINE117
(
  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_117
INDEX IN ts_orderline_117
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 100573 ENDING AT 101439,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE118;
CREATE TABLE ORDER_LINE118
(
  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_118
INDEX IN ts_orderline_118
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 101440 ENDING AT 102306,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE119;
CREATE TABLE ORDER_LINE119

```

```

(
  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_119
INDEX IN ts_orderline_119
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 102307 ENDING AT 103173,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE120;
CREATE TABLE ORDER_LINE120

```

```

(
  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_120
INDEX IN ts_orderline_120
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 103174 ENDING AT 104040,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3741,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

ALLOW OVERFLOW;
connect reset;

```

## DDL/CRTB STOCK.ddl

```

connect to TPCC in share mode;
DROP TABLE STOCK1;
CREATE TABLE STOCK1

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,

```

```

S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_001
INDEX IN ts_stock_001
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 1 ENDING AT 867
)

```

```

ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK2;
CREATE TABLE STOCK2

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_002
INDEX IN ts_stock_002
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 868 ENDING AT 1734
)

```

```

ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK3;
CREATE TABLE STOCK3

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,

```

```

S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_003
INDEX IN ts_stock_003
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 1735 ENDING AT 2601
)

```

```

ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK4;
CREATE TABLE STOCK4

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_004
INDEX IN ts_stock_004
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 2602 ENDING AT 3468
)

```

```

ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK5;
CREATE TABLE STOCK5

```

```

(
  S_REMOTE_CNT INTEGER NOT NULL,
  S_QUANTITY INTEGER NOT NULL,
  S_ORDER_CNT INTEGER NOT NULL,
  S_YTD INTEGER NOT NULL,
  S_DATA VARCHAR(50) NOT NULL,
  S_DIST_01 CHAR(24) NOT NULL,
  S_DIST_02 CHAR(24) NOT NULL,
  S_DIST_03 CHAR(24) NOT NULL,
  S_DIST_04 CHAR(24) NOT NULL,
  S_DIST_05 CHAR(24) NOT NULL,
  S_DIST_06 CHAR(24) NOT NULL,
  S_DIST_07 CHAR(24) NOT NULL,
  S_DIST_08 CHAR(24) NOT NULL,
  S_DIST_09 CHAR(24) NOT NULL,
  S_DIST_10 CHAR(24) NOT NULL,
  S_I_ID INTEGER NOT NULL,
  S_W_ID INTEGER NOT NULL
)

```

```

IN ts_stock_005
INDEX IN ts_stock_005
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 3469 ENDING AT 4335
)

```

```

ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK6;
CREATE TABLE STOCK6

```













```

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 35548 ENDING AT 36414
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK43;
CREATE TABLE STOCK43
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_043
INDEX IN ts_stock_043
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 36415 ENDING AT 37281
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK44;
CREATE TABLE STOCK44
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_044
INDEX IN ts_stock_044
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 37282 ENDING AT 38148
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE STOCK45;
CREATE TABLE STOCK45
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_045
INDEX IN ts_stock_045
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 38149 ENDING AT 39015
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK46;
CREATE TABLE STOCK46
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_046
INDEX IN ts_stock_046
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 39016 ENDING AT 39882
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK47;
CREATE TABLE STOCK47
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,

```

```

S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_047
INDEX IN ts_stock_047
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 39883 ENDING AT 40749
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK48;
CREATE TABLE STOCK48
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_048
INDEX IN ts_stock_048
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 40750 ENDING AT 41616
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK49;
CREATE TABLE STOCK49
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_049
INDEX IN ts_stock_049
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 41617 ENDING AT 42483

```





```

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_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_064
INDEX IN ts_stock_064
ORGANIZE BY KEY SEQUENCE (
S_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 54622 ENDING AT 55488
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK65;
CREATE TABLE STOCK65
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_065
INDEX IN ts_stock_065
ORGANIZE BY KEY SEQUENCE (
S_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 55489 ENDING AT 56355
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK66;
CREATE TABLE STOCK66
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_066
INDEX IN ts_stock_066
ORGANIZE BY KEY SEQUENCE (
S_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 56356 ENDING AT 57222
)
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK67;
CREATE TABLE STOCK67
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_067
INDEX IN ts_stock_067
ORGANIZE BY KEY SEQUENCE (
S_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 57223 ENDING AT 58089
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK68;
CREATE TABLE STOCK68
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_068
INDEX IN ts_stock_068
ORGANIZE BY KEY SEQUENCE (
S_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 58090 ENDING AT 58956
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK69;
CREATE TABLE STOCK69
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,

```

```

S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_069
INDEX IN ts_stock_069
ORGANIZE BY KEY SEQUENCE (
S_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 58957 ENDING AT 59823
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK70;
CREATE TABLE STOCK70
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_070
INDEX IN ts_stock_070
ORGANIZE BY KEY SEQUENCE (
S_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 59824 ENDING AT 60690
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK71;
CREATE TABLE STOCK71
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD INTEGER NOT NULL,
S_DATA VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24) NOT NULL,
S_DIST_02 CHAR(24) NOT NULL,
S_DIST_03 CHAR(24) NOT NULL,
S_DIST_04 CHAR(24) NOT NULL,
S_DIST_05 CHAR(24) NOT NULL,
S_DIST_06 CHAR(24) NOT NULL,
S_DIST_07 CHAR(24) NOT NULL,
S_DIST_08 CHAR(24) NOT NULL,
S_DIST_09 CHAR(24) NOT NULL,
S_DIST_10 CHAR(24) NOT NULL,
S_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_071
INDEX IN ts_stock_071

```









```

)
IN ts_stock_093
INDEX IN ts_stock_093
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 79765 ENDING AT 80631
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK94;
CREATE TABLE STOCK94
(
  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_094
INDEX IN ts_stock_094
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 80632 ENDING AT 81498
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK95;
CREATE TABLE STOCK95
(
  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_095
INDEX IN ts_stock_095
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 81499 ENDING AT 82365
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK96;
CREATE TABLE STOCK96
(

```

```

  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_096
INDEX IN ts_stock_096
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 82366 ENDING AT 83232
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK97;
CREATE TABLE STOCK97
(
  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_097
INDEX IN ts_stock_097
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 83233 ENDING AT 84099
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK98;
CREATE TABLE STOCK98
(
  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_098
INDEX IN ts_stock_098
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 84100 ENDING AT 84966
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK99;
CREATE TABLE STOCK99
(
  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_099
INDEX IN ts_stock_099
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 84967 ENDING AT 85833
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK100;
CREATE TABLE STOCK100
(
  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_100
INDEX IN ts_stock_100
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 85834 ENDING AT 86700
)
ALLOW OVERFLOW;
connect reset;

```





```

S_DIST_10 CHAR(24) NOT NULL,
S_I_ID INTEGER NOT NULL,
S_W_ID INTEGER NOT NULL
)
IN ts_stock_115
INDEX IN ts_stock_115
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 98839 ENDING AT 99705
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK116;
CREATE TABLE STOCK116
(
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_116
INDEX IN ts_stock_116
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 99706 ENDING AT 100572
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK117;
CREATE TABLE STOCK117
(
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_117
INDEX IN ts_stock_117
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 100573 ENDING AT 101439
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE STOCK118;
CREATE TABLE STOCK118
(
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_118
INDEX IN ts_stock_118
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 101440 ENDING AT 102306
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK119;
CREATE TABLE STOCK119
(
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_119
INDEX IN ts_stock_119
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 102307 ENDING AT 103173
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK120;
CREATE TABLE STOCK120
(
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_120
INDEX IN ts_stock_120
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 103174 ENDING AT 104040
)
ALLOW OVERFLOW;
connect reset;
DDL/CRTB_WAREHOUSE.ddl
connect to TPCC in share mode;
DROP TABLE WAREHOUSE1;
CREATE TABLE WAREHOUSE1
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_001
INDEX IN ts_ware_001
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 1 ENDING AT 867
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE2;
CREATE TABLE WAREHOUSE2
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_002
INDEX IN ts_ware_002
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 868 ENDING AT 1734
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE3;
CREATE TABLE WAREHOUSE3
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,

```

```

W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_003
INDEX IN ts_ware_003
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 1735 ENDING AT 2601
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE4;
CREATE TABLE WAREHOUSE4
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_004
INDEX IN ts_ware_004
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 2602 ENDING AT 3468
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE5;
CREATE TABLE WAREHOUSE5
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_005
INDEX IN ts_ware_005
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 3469 ENDING AT 4335
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE6;
CREATE TABLE WAREHOUSE6
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_006
INDEX IN ts_ware_006
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 4336 ENDING AT 5202
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE7;
CREATE TABLE WAREHOUSE7
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_007
INDEX IN ts_ware_007
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 5203 ENDING AT 6069
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE8;
CREATE TABLE WAREHOUSE8
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_008
INDEX IN ts_ware_008
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 6070 ENDING AT 6936
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE9;
CREATE TABLE WAREHOUSE9
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_009
INDEX IN ts_ware_009
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 6937 ENDING AT 7803
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE10;
CREATE TABLE WAREHOUSE10
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,

```

```

W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_010
INDEX IN ts_ware_010
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 7804 ENDING AT 8670
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE11;
CREATE TABLE WAREHOUSE11
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_011
INDEX IN ts_ware_011
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 8671 ENDING AT 9537
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE12;
CREATE TABLE WAREHOUSE12
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_012
INDEX IN ts_ware_012
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 9538 ENDING AT 10404
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE13;
CREATE TABLE WAREHOUSE13
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_013
INDEX IN ts_ware_013
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 10405 ENDING AT 11271
)

```



```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE14;
CREATE TABLE WAREHOUSE14
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_014
INDEX IN ts_ware_014
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 11272 ENDING AT 12138
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE15;
CREATE TABLE WAREHOUSE15
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_015
INDEX IN ts_ware_015
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 12139 ENDING AT 13005
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE16;
CREATE TABLE WAREHOUSE16
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_016
INDEX IN ts_ware_016
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 13006 ENDING AT 13872
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE17;
CREATE TABLE WAREHOUSE17
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,

```

```

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_017
INDEX IN ts_ware_017
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 13873 ENDING AT 14739
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE18;
CREATE TABLE WAREHOUSE18
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_018
INDEX IN ts_ware_018
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 14740 ENDING AT 15606
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE19;
CREATE TABLE WAREHOUSE19
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_019
INDEX IN ts_ware_019
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 15607 ENDING AT 16473
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE20;
CREATE TABLE WAREHOUSE20
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_020
INDEX IN ts_ware_020
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 16474 ENDING AT 17340
)

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE21;
CREATE TABLE WAREHOUSE21
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_021
INDEX IN ts_ware_021
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 17341 ENDING AT 18207
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE22;
CREATE TABLE WAREHOUSE22
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_022
INDEX IN ts_ware_022
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 18208 ENDING AT 19074
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE23;
CREATE TABLE WAREHOUSE23
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_023
INDEX IN ts_ware_023
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 19075 ENDING AT 19941
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE24;
CREATE TABLE WAREHOUSE24
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,

```

```

W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_024
INDEX IN ts_ware_024
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 19942 ENDING AT 20808
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE25;
CREATE TABLE WAREHOUSE25
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_025
INDEX IN ts_ware_025
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 20809 ENDING AT 21675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE26;
CREATE TABLE WAREHOUSE26
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_026
INDEX IN ts_ware_026
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 21676 ENDING AT 22542
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE27;
CREATE TABLE WAREHOUSE27
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_027
INDEX IN ts_ware_027
ORGANIZE BY KEY SEQUENCE (

```

```

W_ID STARTING FROM 22543 ENDING AT 23409
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE28;
CREATE TABLE WAREHOUSE28
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_028
INDEX IN ts_ware_028
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 23410 ENDING AT 24276
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE29;
CREATE TABLE WAREHOUSE29
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_029
INDEX IN ts_ware_029
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 24277 ENDING AT 25143
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE30;
CREATE TABLE WAREHOUSE30
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_030
INDEX IN ts_ware_030
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 25144 ENDING AT 26010
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE31;
CREATE TABLE WAREHOUSE31
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,

```

```

W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
)
IN ts_ware_031
INDEX IN ts_ware_031
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 26011 ENDING AT 26877
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE32;
CREATE TABLE WAREHOUSE32
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_032
INDEX IN ts_ware_032
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 26878 ENDING AT 27744
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE33;
CREATE TABLE WAREHOUSE33
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_033
INDEX IN ts_ware_033
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 27745 ENDING AT 28611
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE34;
CREATE TABLE WAREHOUSE34
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_034
INDEX IN ts_ware_034

```

```

ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 28612 ENDING AT 29478
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE35;
CREATE TABLE WAREHOUSE35
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_035
INDEX IN ts_ware_035
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 29479 ENDING AT 30345
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE36;
CREATE TABLE WAREHOUSE36
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_036
INDEX IN ts_ware_036
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 30346 ENDING AT 31212
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE37;
CREATE TABLE WAREHOUSE37
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_037
INDEX IN ts_ware_037
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 31213 ENDING AT 32079
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE38;
CREATE TABLE WAREHOUSE38
(
  W_NAME CHAR(10) NOT NULL,

```

```

  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_038
INDEX IN ts_ware_038
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 32080 ENDING AT 32946
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE39;
CREATE TABLE WAREHOUSE39
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_039
INDEX IN ts_ware_039
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 32947 ENDING AT 33813
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE40;
CREATE TABLE WAREHOUSE40
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_040
INDEX IN ts_ware_040
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 33814 ENDING AT 34680
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE41;
CREATE TABLE WAREHOUSE41
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_041

```

```

INDEX IN ts_ware_041
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 34681 ENDING AT 35547
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE42;
CREATE TABLE WAREHOUSE42
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_042
INDEX IN ts_ware_042
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 35548 ENDING AT 36414
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE43;
CREATE TABLE WAREHOUSE43
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_043
INDEX IN ts_ware_043
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 36415 ENDING AT 37281
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE44;
CREATE TABLE WAREHOUSE44
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_044
INDEX IN ts_ware_044
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 37282 ENDING AT 38148
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE45;
CREATE TABLE WAREHOUSE45
(

```

```

W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_045
INDEX IN ts_ware_045
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 38149 ENDING AT 39015
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE46;
CREATE TABLE WAREHOUSE46
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_046
INDEX IN ts_ware_046
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 39016 ENDING AT 39882
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE47;
CREATE TABLE WAREHOUSE47
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_047
INDEX IN ts_ware_047
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 39883 ENDING AT 40749
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE48;
CREATE TABLE WAREHOUSE48
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
)

```

```

IN ts_ware_048
INDEX IN ts_ware_048
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 40750 ENDING AT 41616
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE49;
CREATE TABLE WAREHOUSE49
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_049
INDEX IN ts_ware_049
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 41617 ENDING AT 42483
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE50;
CREATE TABLE WAREHOUSE50
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_050
INDEX IN ts_ware_050
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 42484 ENDING AT 43350
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE51;
CREATE TABLE WAREHOUSE51
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_051
INDEX IN ts_ware_051
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 43351 ENDING AT 44217
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE52;
CREATE TABLE WAREHOUSE52

```

```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
)
IN ts_ware_052
INDEX IN ts_ware_052
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 44218 ENDING AT 45084
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE53;
CREATE TABLE WAREHOUSE53
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
)
IN ts_ware_053
INDEX IN ts_ware_053
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 45085 ENDING AT 45951
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE54;
CREATE TABLE WAREHOUSE54
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
)
IN ts_ware_054
INDEX IN ts_ware_054
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 45952 ENDING AT 46818
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE55;
CREATE TABLE WAREHOUSE55
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
)

```

```

)
IN ts_ware_055
INDEX IN ts_ware_055
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 46819 ENDING AT 47685
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE56;
CREATE TABLE WAREHOUSE56
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_056
INDEX IN ts_ware_056
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 47686 ENDING AT 48552
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE57;
CREATE TABLE WAREHOUSE57
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_057
INDEX IN ts_ware_057
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 48553 ENDING AT 49419
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE58;
CREATE TABLE WAREHOUSE58
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_058
INDEX IN ts_ware_058
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 49420 ENDING AT 50286
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE59;

```

```

CREATE TABLE WAREHOUSE59
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_059
INDEX IN ts_ware_059
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 50287 ENDING AT 51153
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE60;
CREATE TABLE WAREHOUSE60
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_060
INDEX IN ts_ware_060
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 51154 ENDING AT 52020
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE61;
CREATE TABLE WAREHOUSE61
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_061
INDEX IN ts_ware_061
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 52021 ENDING AT 52887
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE62;
CREATE TABLE WAREHOUSE62
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,

```

```

  W_ID INTEGER NOT NULL
)
IN ts_ware_062
INDEX IN ts_ware_062
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 52888 ENDING AT 53754
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE63;
CREATE TABLE WAREHOUSE63
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_063
INDEX IN ts_ware_063
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 53755 ENDING AT 54621
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE64;
CREATE TABLE WAREHOUSE64
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_064
INDEX IN ts_ware_064
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 54622 ENDING AT 55488
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE65;
CREATE TABLE WAREHOUSE65
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_065
INDEX IN ts_ware_065
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 55489 ENDING AT 56355
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE WAREHOUSE66;
CREATE TABLE WAREHOUSE66
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_066
INDEX IN ts_ware_066
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 56356 ENDING AT 57222
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE67;
CREATE TABLE WAREHOUSE67
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_067
INDEX IN ts_ware_067
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 57223 ENDING AT 58089
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE68;
CREATE TABLE WAREHOUSE68
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_068
INDEX IN ts_ware_068
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 58090 ENDING AT 58956
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE69;
CREATE TABLE WAREHOUSE69
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,

```

```

  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_069
INDEX IN ts_ware_069
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 58957 ENDING AT 59823
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE70;
CREATE TABLE WAREHOUSE70
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_070
INDEX IN ts_ware_070
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 59824 ENDING AT 60690
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE71;
CREATE TABLE WAREHOUSE71
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_071
INDEX IN ts_ware_071
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 60691 ENDING AT 61557
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE72;
CREATE TABLE WAREHOUSE72
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_072
INDEX IN ts_ware_072
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 61558 ENDING AT 62424
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE WAREHOUSE73;
CREATE TABLE WAREHOUSE73
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_073
INDEX IN ts_ware_073
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 62425 ENDING AT 63291
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE74;
CREATE TABLE WAREHOUSE74
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_074
INDEX IN ts_ware_074
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 63292 ENDING AT 64158
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE75;
CREATE TABLE WAREHOUSE75
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX REAL NOT NULL,
  W_YTD DECIMAL(12,2) NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_075
INDEX IN ts_ware_075
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 64159 ENDING AT 65025
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE76;
CREATE TABLE WAREHOUSE76
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,

```

```

W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_076
INDEX IN ts_ware_076
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 65026 ENDING AT 65892
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE77;
CREATE TABLE WAREHOUSE77
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_077
INDEX IN ts_ware_077
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 65893 ENDING AT 66759
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE78;
CREATE TABLE WAREHOUSE78
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_078
INDEX IN ts_ware_078
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 66760 ENDING AT 67626
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE79;
CREATE TABLE WAREHOUSE79
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_079
INDEX IN ts_ware_079
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 67627 ENDING AT 68493
)
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE80;
CREATE TABLE WAREHOUSE80
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX REAL NOT NULL,
W_YTD DECIMAL(12,2) NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_080
INDEX IN ts_ware_080
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 68494 ENDING AT 69360
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE81;
CREATE TABLE WAREHOUSE81
(
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_ware_081
INDEX IN ts_ware_081
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 69361 ENDING AT 70227
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE82;
CREATE TABLE WAREHOUSE82
(
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_ware_082
INDEX IN ts_ware_082
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 70228 ENDING AT 71094
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE83;
CREATE TABLE WAREHOUSE83
(
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_ware_083
INDEX IN ts_ware_083
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 71095 ENDING AT 71961
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE84;
CREATE TABLE WAREHOUSE84
(
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_ware_084
INDEX IN ts_ware_084
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 71962 ENDING AT 72828
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE85;
CREATE TABLE WAREHOUSE85
(
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_ware_085
INDEX IN ts_ware_085
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 72829 ENDING AT 73695
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE86;
CREATE TABLE WAREHOUSE86
(
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_ware_086
INDEX IN ts_ware_086
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 73696 ENDING AT 74562
)
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE87;
CREATE TABLE WAREHOUSE87
(
  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_ware_087
INDEX IN ts_ware_087
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 74563 ENDING AT 75429
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE88;
CREATE TABLE WAREHOUSE88
(
  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_ware_088
INDEX IN ts_ware_088
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 75430 ENDING AT 76296
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE89;
CREATE TABLE WAREHOUSE89
(
  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_ware_089
INDEX IN ts_ware_089
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 76297 ENDING AT 77163
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE90;
CREATE TABLE WAREHOUSE90
(
  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_ware_090
INDEX IN ts_ware_090
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 77164 ENDING AT 78030
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE91;
CREATE TABLE WAREHOUSE91
(
  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_ware_091
INDEX IN ts_ware_091
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 78031 ENDING AT 78897
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE92;
CREATE TABLE WAREHOUSE92
(
  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_ware_092
INDEX IN ts_ware_092
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 78898 ENDING AT 79764
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE93;
CREATE TABLE WAREHOUSE93
(
  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_ware_093
INDEX IN ts_ware_093
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 79765 ENDING AT 80631
)

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE94;
CREATE TABLE WAREHOUSE94
(
  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_ware_094
INDEX IN ts_ware_094
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 80632 ENDING AT 81498
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE95;
CREATE TABLE WAREHOUSE95
(
  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_ware_095
INDEX IN ts_ware_095
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 81499 ENDING AT 82365
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE96;
CREATE TABLE WAREHOUSE96
(
  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_ware_096
INDEX IN ts_ware_096
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 82366 ENDING AT 83232
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE97;
CREATE TABLE WAREHOUSE97
(
  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_ware_097
INDEX IN ts_ware_097
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 83233 ENDING AT 84099
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE98;
CREATE TABLE WAREHOUSE98
(
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_ware_098
INDEX IN ts_ware_098
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 84100 ENDING AT 84966
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE99;
CREATE TABLE WAREHOUSE99
(
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_ware_099
INDEX IN ts_ware_099
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 84967 ENDING AT 85833
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE100;
CREATE TABLE WAREHOUSE100
(
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_ware_100
INDEX IN ts_ware_100
ORGANIZE BY KEY SEQUENCE (

```

```

W_ID STARTING FROM 85834 ENDING AT 86700
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE101;
CREATE TABLE WAREHOUSE101
(
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_ware_101
INDEX IN ts_ware_101
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 86701 ENDING AT 87567
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE102;
CREATE TABLE WAREHOUSE102
(
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_ware_102
INDEX IN ts_ware_102
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 87568 ENDING AT 88434
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE103;
CREATE TABLE WAREHOUSE103
(
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_ware_103
INDEX IN ts_ware_103
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 88435 ENDING AT 89301
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE104;
CREATE TABLE WAREHOUSE104
(
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_ware_104
INDEX IN ts_ware_104
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 89302 ENDING AT 90168
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE105;
CREATE TABLE WAREHOUSE105
(
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_ware_105
INDEX IN ts_ware_105
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 90169 ENDING AT 91035
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE106;
CREATE TABLE WAREHOUSE106
(
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_ware_106
INDEX IN ts_ware_106
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 91036 ENDING AT 91902
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE107;
CREATE TABLE WAREHOUSE107
(
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_ware_107
INDEX IN ts_ware_107

```

```

ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 91903 ENDING AT 92769
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE108;
CREATE TABLE WAREHOUSE108
(
  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_ware_108
INDEX IN ts_ware_108
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 92770 ENDING AT 93636
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE109;
CREATE TABLE WAREHOUSE109
(
  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_ware_109
INDEX IN ts_ware_109
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 93637 ENDING AT 94503
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE110;
CREATE TABLE WAREHOUSE110
(
  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_ware_110
INDEX IN ts_ware_110
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 94504 ENDING AT 95370
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE111;
CREATE TABLE WAREHOUSE111
(
  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_ware_111
INDEX IN ts_ware_111
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 95371 ENDING AT 96237
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE112;
CREATE TABLE WAREHOUSE112
(
  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_ware_112
INDEX IN ts_ware_112
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 96238 ENDING AT 97104
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE113;
CREATE TABLE WAREHOUSE113
(
  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_ware_113
INDEX IN ts_ware_113
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 97105 ENDING AT 97971
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE114;
CREATE TABLE WAREHOUSE114
(
  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_ware_114

```

```

INDEX IN ts_ware_114
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 97972 ENDING AT 98838
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE115;
CREATE TABLE WAREHOUSE115
(
  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_ware_115
INDEX IN ts_ware_115
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 98839 ENDING AT 99705
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE116;
CREATE TABLE WAREHOUSE116
(
  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_ware_116
INDEX IN ts_ware_116
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 99706 ENDING AT 100572
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE117;
CREATE TABLE WAREHOUSE117
(
  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_ware_117
INDEX IN ts_ware_117
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 100573 ENDING AT 101439
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE118;
CREATE TABLE WAREHOUSE118
(

```

```

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_ware_118
INDEX IN ts_ware_118
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 101440 ENDING AT 102306
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE119;
CREATE TABLE WAREHOUSE119
(
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_ware_119
INDEX IN ts_ware_119
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 102307 ENDING AT 103173
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE120;
CREATE TABLE WAREHOUSE120
(
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_ware_120
INDEX IN ts_ware_120
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 103174 ENDING AT 104040
)
)
ALLOW OVERFLOW;
connect reset;

```

## DDL/CRVW CUSTOMER.ddl

```

connect to TPCC in share mode;
DROP VIEW CUSTOMER;
CREATE VIEW CUSTOMER
(C_ID,
C_STATE,
C_ZIP,
C_PHONE,
C_SINCE,
C_CREDIT_LIM,

```

```

C_MIDDLE,
C_CREDIT,
C_DISCOUNT,
C_DATA,
C_LAST,
C_FIRST,
C_STREET_1,
C_STREET_2,
C_CITY,
C_D_ID,
C_W_ID,
C_DELIVERY_CNT,
C_BALANCE,
C_YTD_PAYMENT,
C_PAYMENT_CNT
) AS SELECT * FROM CUSTOMER1 UNION ALL
SELECT * FROM CUSTOMER2 UNION ALL
SELECT * FROM CUSTOMER3 UNION ALL
SELECT * FROM CUSTOMER4 UNION ALL
SELECT * FROM CUSTOMER5 UNION ALL
SELECT * FROM CUSTOMER6 UNION ALL
SELECT * FROM CUSTOMER7 UNION ALL
SELECT * FROM CUSTOMER8 UNION ALL
SELECT * FROM CUSTOMER9 UNION ALL
SELECT * FROM CUSTOMER10 UNION ALL
SELECT * FROM CUSTOMER11 UNION ALL
SELECT * FROM CUSTOMER12 UNION ALL
SELECT * FROM CUSTOMER13 UNION ALL
SELECT * FROM CUSTOMER14 UNION ALL
SELECT * FROM CUSTOMER15 UNION ALL
SELECT * FROM CUSTOMER16 UNION ALL
SELECT * FROM CUSTOMER17 UNION ALL
SELECT * FROM CUSTOMER18 UNION ALL
SELECT * FROM CUSTOMER19 UNION ALL
SELECT * FROM CUSTOMER20 UNION ALL
SELECT * FROM CUSTOMER21 UNION ALL
SELECT * FROM CUSTOMER22 UNION ALL
SELECT * FROM CUSTOMER23 UNION ALL
SELECT * FROM CUSTOMER24 UNION ALL
SELECT * FROM CUSTOMER25 UNION ALL
SELECT * FROM CUSTOMER26 UNION ALL
SELECT * FROM CUSTOMER27 UNION ALL
SELECT * FROM CUSTOMER28 UNION ALL
SELECT * FROM CUSTOMER29 UNION ALL
SELECT * FROM CUSTOMER30 UNION ALL
SELECT * FROM CUSTOMER31 UNION ALL
SELECT * FROM CUSTOMER32 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38 UNION ALL
SELECT * FROM CUSTOMER39 UNION ALL
SELECT * FROM CUSTOMER40 UNION ALL
SELECT * FROM CUSTOMER41 UNION ALL
SELECT * FROM CUSTOMER42 UNION ALL
SELECT * FROM CUSTOMER43 UNION ALL
SELECT * FROM CUSTOMER44 UNION ALL
SELECT * FROM CUSTOMER45 UNION ALL
SELECT * FROM CUSTOMER46 UNION ALL
SELECT * FROM CUSTOMER47 UNION ALL
SELECT * FROM CUSTOMER48 UNION ALL
SELECT * FROM CUSTOMER49 UNION ALL
SELECT * FROM CUSTOMER50 UNION ALL
SELECT * FROM CUSTOMER51 UNION ALL
SELECT * FROM CUSTOMER52 UNION ALL
SELECT * FROM CUSTOMER53 UNION ALL
SELECT * FROM CUSTOMER54 UNION ALL
SELECT * FROM CUSTOMER55 UNION ALL
SELECT * FROM CUSTOMER56 UNION ALL
SELECT * FROM CUSTOMER57 UNION ALL
SELECT * FROM CUSTOMER58 UNION ALL

```

```

SELECT * FROM CUSTOMER59 UNION ALL
SELECT * FROM CUSTOMER60 UNION ALL
SELECT * FROM CUSTOMER61 UNION ALL
SELECT * FROM CUSTOMER62 UNION ALL
SELECT * FROM CUSTOMER63 UNION ALL
SELECT * FROM CUSTOMER64 UNION ALL
SELECT * FROM CUSTOMER65 UNION ALL
SELECT * FROM CUSTOMER66 UNION ALL
SELECT * FROM CUSTOMER67 UNION ALL
SELECT * FROM CUSTOMER68 UNION ALL
SELECT * FROM CUSTOMER69 UNION ALL
SELECT * FROM CUSTOMER70 UNION ALL
SELECT * FROM CUSTOMER71 UNION ALL
SELECT * FROM CUSTOMER72 UNION ALL
SELECT * FROM CUSTOMER73 UNION ALL
SELECT * FROM CUSTOMER74 UNION ALL
SELECT * FROM CUSTOMER75 UNION ALL
SELECT * FROM CUSTOMER76 UNION ALL
SELECT * FROM CUSTOMER77 UNION ALL
SELECT * FROM CUSTOMER78 UNION ALL
SELECT * FROM CUSTOMER79 UNION ALL
SELECT * FROM CUSTOMER80 UNION ALL
SELECT * FROM CUSTOMER81 UNION ALL
SELECT * FROM CUSTOMER82 UNION ALL
SELECT * FROM CUSTOMER83 UNION ALL
SELECT * FROM CUSTOMER84 UNION ALL
SELECT * FROM CUSTOMER85 UNION ALL
SELECT * FROM CUSTOMER86 UNION ALL
SELECT * FROM CUSTOMER87 UNION ALL
SELECT * FROM CUSTOMER88 UNION ALL
SELECT * FROM CUSTOMER89 UNION ALL
SELECT * FROM CUSTOMER90 UNION ALL
SELECT * FROM CUSTOMER91 UNION ALL
SELECT * FROM CUSTOMER92 UNION ALL
SELECT * FROM CUSTOMER93 UNION ALL
SELECT * FROM CUSTOMER94 UNION ALL
SELECT * FROM CUSTOMER95 UNION ALL
SELECT * FROM CUSTOMER96 UNION ALL
SELECT * FROM CUSTOMER97 UNION ALL
SELECT * FROM CUSTOMER98 UNION ALL
SELECT * FROM CUSTOMER99 UNION ALL
SELECT * FROM CUSTOMER100 UNION ALL
SELECT * FROM CUSTOMER101 UNION ALL
SELECT * FROM CUSTOMER102 UNION ALL
SELECT * FROM CUSTOMER103 UNION ALL
SELECT * FROM CUSTOMER104 UNION ALL
SELECT * FROM CUSTOMER105 UNION ALL
SELECT * FROM CUSTOMER106 UNION ALL
SELECT * FROM CUSTOMER107 UNION ALL
SELECT * FROM CUSTOMER108 UNION ALL
SELECT * FROM CUSTOMER109 UNION ALL
SELECT * FROM CUSTOMER110 UNION ALL
SELECT * FROM CUSTOMER111 UNION ALL
SELECT * FROM CUSTOMER112 UNION ALL
SELECT * FROM CUSTOMER113 UNION ALL
SELECT * FROM CUSTOMER114 UNION ALL
SELECT * FROM CUSTOMER115 UNION ALL
SELECT * FROM CUSTOMER116 UNION ALL
SELECT * FROM CUSTOMER117 UNION ALL
SELECT * FROM CUSTOMER118 UNION ALL
SELECT * FROM CUSTOMER119 UNION ALL
SELECT * FROM CUSTOMER120
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

## DDL/CRVW DISTRICT.ddl

```

connect to TPCC in share mode;
DROP VIEW DISTRICT;
CREATE VIEW DISTRICT

```







```

SELECT * FROM ORDERS115 UNION ALL
SELECT * FROM ORDERS116 UNION ALL
SELECT * FROM ORDERS117 UNION ALL
SELECT * FROM ORDERS118 UNION ALL
SELECT * FROM ORDERS119 UNION ALL
SELECT * FROM ORDERS120
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

## DDL/CRVW ORDER LINE.ddl

```

connect to TPCC in share mode;
DROP VIEW ORDER_LINE;
CREATE VIEW ORDER_LINE
(OL_DELIVERY_D,
OL_AMOUNT,
OL_I_ID,
OL_SUPPLY_W_ID,
OL_QUANTITY,
OL_DIST_INFO,
OL_O_ID,
OL_D_ID,
OL_W_ID,
OL_NUMBER
) AS SELECT * FROM ORDER_LINE1 UNION ALL
SELECT * FROM ORDER_LINE2 UNION ALL
SELECT * FROM ORDER_LINE3 UNION ALL
SELECT * FROM ORDER_LINE4 UNION ALL
SELECT * FROM ORDER_LINE5 UNION ALL
SELECT * FROM ORDER_LINE6 UNION ALL
SELECT * FROM ORDER_LINE7 UNION ALL
SELECT * FROM ORDER_LINE8 UNION ALL
SELECT * FROM ORDER_LINE9 UNION ALL
SELECT * FROM ORDER_LINE10 UNION ALL
SELECT * FROM ORDER_LINE11 UNION ALL
SELECT * FROM ORDER_LINE12 UNION ALL
SELECT * FROM ORDER_LINE13 UNION ALL
SELECT * FROM ORDER_LINE14 UNION ALL
SELECT * FROM ORDER_LINE15 UNION ALL
SELECT * FROM ORDER_LINE16 UNION ALL
SELECT * FROM ORDER_LINE17 UNION ALL
SELECT * FROM ORDER_LINE18 UNION ALL
SELECT * FROM ORDER_LINE19 UNION ALL
SELECT * FROM ORDER_LINE20 UNION ALL
SELECT * FROM ORDER_LINE21 UNION ALL
SELECT * FROM ORDER_LINE22 UNION ALL
SELECT * FROM ORDER_LINE23 UNION ALL
SELECT * FROM ORDER_LINE24 UNION ALL
SELECT * FROM ORDER_LINE25 UNION ALL
SELECT * FROM ORDER_LINE26 UNION ALL
SELECT * FROM ORDER_LINE27 UNION ALL
SELECT * FROM ORDER_LINE28 UNION ALL
SELECT * FROM ORDER_LINE29 UNION ALL
SELECT * FROM ORDER_LINE30 UNION ALL
SELECT * FROM ORDER_LINE31 UNION ALL
SELECT * FROM ORDER_LINE32 UNION ALL
SELECT * FROM ORDER_LINE33 UNION ALL
SELECT * FROM ORDER_LINE34 UNION ALL
SELECT * FROM ORDER_LINE35 UNION ALL
SELECT * FROM ORDER_LINE36 UNION ALL
SELECT * FROM ORDER_LINE37 UNION ALL
SELECT * FROM ORDER_LINE38 UNION ALL
SELECT * FROM ORDER_LINE39 UNION ALL
SELECT * FROM ORDER_LINE40 UNION ALL
SELECT * FROM ORDER_LINE41 UNION ALL
SELECT * FROM ORDER_LINE42 UNION ALL
SELECT * FROM ORDER_LINE43 UNION ALL
SELECT * FROM ORDER_LINE44 UNION ALL
SELECT * FROM ORDER_LINE45 UNION ALL
SELECT * FROM ORDER_LINE46 UNION ALL

```

```

SELECT * FROM ORDER_LINE47 UNION ALL
SELECT * FROM ORDER_LINE48 UNION ALL
SELECT * FROM ORDER_LINE49 UNION ALL
SELECT * FROM ORDER_LINE50 UNION ALL
SELECT * FROM ORDER_LINE51 UNION ALL
SELECT * FROM ORDER_LINE52 UNION ALL
SELECT * FROM ORDER_LINE53 UNION ALL
SELECT * FROM ORDER_LINE54 UNION ALL
SELECT * FROM ORDER_LINE55 UNION ALL
SELECT * FROM ORDER_LINE56 UNION ALL
SELECT * FROM ORDER_LINE57 UNION ALL
SELECT * FROM ORDER_LINE58 UNION ALL
SELECT * FROM ORDER_LINE59 UNION ALL
SELECT * FROM ORDER_LINE60 UNION ALL
SELECT * FROM ORDER_LINE61 UNION ALL
SELECT * FROM ORDER_LINE62 UNION ALL
SELECT * FROM ORDER_LINE63 UNION ALL
SELECT * FROM ORDER_LINE64 UNION ALL
SELECT * FROM ORDER_LINE65 UNION ALL
SELECT * FROM ORDER_LINE66 UNION ALL
SELECT * FROM ORDER_LINE67 UNION ALL
SELECT * FROM ORDER_LINE68 UNION ALL
SELECT * FROM ORDER_LINE69 UNION ALL
SELECT * FROM ORDER_LINE70 UNION ALL
SELECT * FROM ORDER_LINE71 UNION ALL
SELECT * FROM ORDER_LINE72 UNION ALL
SELECT * FROM ORDER_LINE73 UNION ALL
SELECT * FROM ORDER_LINE74 UNION ALL
SELECT * FROM ORDER_LINE75 UNION ALL
SELECT * FROM ORDER_LINE76 UNION ALL
SELECT * FROM ORDER_LINE77 UNION ALL
SELECT * FROM ORDER_LINE78 UNION ALL
SELECT * FROM ORDER_LINE79 UNION ALL
SELECT * FROM ORDER_LINE80 UNION ALL
SELECT * FROM ORDER_LINE81 UNION ALL
SELECT * FROM ORDER_LINE82 UNION ALL
SELECT * FROM ORDER_LINE83 UNION ALL
SELECT * FROM ORDER_LINE84 UNION ALL
SELECT * FROM ORDER_LINE85 UNION ALL
SELECT * FROM ORDER_LINE86 UNION ALL
SELECT * FROM ORDER_LINE87 UNION ALL
SELECT * FROM ORDER_LINE88 UNION ALL
SELECT * FROM ORDER_LINE89 UNION ALL
SELECT * FROM ORDER_LINE90 UNION ALL
SELECT * FROM ORDER_LINE91 UNION ALL
SELECT * FROM ORDER_LINE92 UNION ALL
SELECT * FROM ORDER_LINE93 UNION ALL
SELECT * FROM ORDER_LINE94 UNION ALL
SELECT * FROM ORDER_LINE95 UNION ALL
SELECT * FROM ORDER_LINE96 UNION ALL
SELECT * FROM ORDER_LINE97 UNION ALL
SELECT * FROM ORDER_LINE98 UNION ALL
SELECT * FROM ORDER_LINE99 UNION ALL
SELECT * FROM ORDER_LINE100 UNION ALL
SELECT * FROM ORDER_LINE101 UNION ALL
SELECT * FROM ORDER_LINE102 UNION ALL
SELECT * FROM ORDER_LINE103 UNION ALL
SELECT * FROM ORDER_LINE104 UNION ALL
SELECT * FROM ORDER_LINE105 UNION ALL
SELECT * FROM ORDER_LINE106 UNION ALL
SELECT * FROM ORDER_LINE107 UNION ALL
SELECT * FROM ORDER_LINE108 UNION ALL
SELECT * FROM ORDER_LINE109 UNION ALL
SELECT * FROM ORDER_LINE110 UNION ALL
SELECT * FROM ORDER_LINE111 UNION ALL
SELECT * FROM ORDER_LINE112 UNION ALL
SELECT * FROM ORDER_LINE113 UNION ALL
SELECT * FROM ORDER_LINE114 UNION ALL
SELECT * FROM ORDER_LINE115 UNION ALL
SELECT * FROM ORDER_LINE116 UNION ALL
SELECT * FROM ORDER_LINE117 UNION ALL
SELECT * FROM ORDER_LINE118 UNION ALL
SELECT * FROM ORDER_LINE119 UNION ALL

```

```

SELECT * FROM ORDER_LINE120
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

## DDL/CRVW STOCK.ddl

```

connect to TPCC in share mode;
DROP VIEW STOCK;
CREATE VIEW STOCK
(S_REMOTE_CNT,
S_QUANTITY,
S_ORDER_CNT,
S_YTD,
S_DATA,
S_DIST_01,
S_DIST_02,
S_DIST_03,
S_DIST_04,
S_DIST_05,
S_DIST_06,
S_DIST_07,
S_DIST_08,
S_DIST_09,
S_DIST_10,
S_I_ID,
S_W_ID
) AS SELECT * FROM STOCK1 UNION ALL
SELECT * FROM STOCK2 UNION ALL
SELECT * FROM STOCK3 UNION ALL
SELECT * FROM STOCK4 UNION ALL
SELECT * FROM STOCK5 UNION ALL
SELECT * FROM STOCK6 UNION ALL
SELECT * FROM STOCK7 UNION ALL
SELECT * FROM STOCK8 UNION ALL
SELECT * FROM STOCK9 UNION ALL
SELECT * FROM STOCK10 UNION ALL
SELECT * FROM STOCK11 UNION ALL
SELECT * FROM STOCK12 UNION ALL
SELECT * FROM STOCK13 UNION ALL
SELECT * FROM STOCK14 UNION ALL
SELECT * FROM STOCK15 UNION ALL
SELECT * FROM STOCK16 UNION ALL
SELECT * FROM STOCK17 UNION ALL
SELECT * FROM STOCK18 UNION ALL
SELECT * FROM STOCK19 UNION ALL
SELECT * FROM STOCK20 UNION ALL
SELECT * FROM STOCK21 UNION ALL
SELECT * FROM STOCK22 UNION ALL
SELECT * FROM STOCK23 UNION ALL
SELECT * FROM STOCK24 UNION ALL
SELECT * FROM STOCK25 UNION ALL
SELECT * FROM STOCK26 UNION ALL
SELECT * FROM STOCK27 UNION ALL
SELECT * FROM STOCK28 UNION ALL
SELECT * FROM STOCK29 UNION ALL
SELECT * FROM STOCK30 UNION ALL
SELECT * FROM STOCK31 UNION ALL
SELECT * FROM STOCK32 UNION ALL
SELECT * FROM STOCK33 UNION ALL
SELECT * FROM STOCK34 UNION ALL
SELECT * FROM STOCK35 UNION ALL
SELECT * FROM STOCK36 UNION ALL
SELECT * FROM STOCK37 UNION ALL
SELECT * FROM STOCK38 UNION ALL
SELECT * FROM STOCK39 UNION ALL
SELECT * FROM STOCK40 UNION ALL
SELECT * FROM STOCK41 UNION ALL
SELECT * FROM STOCK42 UNION ALL
SELECT * FROM STOCK43 UNION ALL
SELECT * FROM STOCK44 UNION ALL

```











/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 68494 69360 -f1  
/flats/F1\_080/history\_080\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 69361 70227 -f1  
/flats/F1\_081/history\_081\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 70228 71094 -f1  
/flats/F1\_082/history\_082\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 71095 71961 -f1  
/flats/F1\_083/history\_083\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 71962 72828 -f1  
/flats/F1\_084/history\_084\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 72829 73695 -f1  
/flats/F1\_085/history\_085\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 73696 74562 -f1  
/flats/F1\_086/history\_086\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 74563 75429 -f1  
/flats/F1\_087/history\_087\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 75430 76296 -f1  
/flats/F1\_088/history\_088\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 76297 77163 -f1  
/flats/F1\_089/history\_089\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 77164 78030 -f1  
/flats/F1\_090/history\_090\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 78031 78897 -f1  
/flats/F1\_091/history\_091\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 78898 79764 -f1  
/flats/F1\_092/history\_092\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 79765 80631 -f1  
/flats/F1\_093/history\_093\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 80632 81498 -f1  
/flats/F1\_094/history\_094\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 81499 82365 -f1  
/flats/F1\_095/history\_095\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 82366 83232 -f1  
/flats/F1\_096/history\_096\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 83233 84099 -f1  
/flats/F1\_097/history\_097\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 84100 84966 -f1  
/flats/F1\_098/history\_098\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 84967 85833 -f1  
/flats/F1\_099/history\_099\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 85834 86700 -f1  
/flats/F1\_100/history\_100\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 86701 87567 -f1  
/flats/F1\_101/history\_101\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 87568 88434 -f1  
/flats/F1\_102/history\_102\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 88435 89301 -f1  
/flats/F1\_103/history\_103\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 89302 90168 -f1  
/flats/F1\_104/history\_104\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 90169 91035 -f1  
/flats/F1\_105/history\_105\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 91036 91902 -f1  
/flats/F1\_106/history\_106\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 91903 92769 -f1  
/flats/F1\_107/history\_107\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 92770 93636 -f1  
/flats/F1\_108/history\_108\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 93637 94503 -f1  
/flats/F1\_109/history\_109\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 94504 95370 -f1  
/flats/F1\_110/history\_110\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 95371 96237 -f1  
/flats/F1\_111/history\_111\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 96238 97104 -f1  
/flats/F1\_112/history\_112\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 97105 97971 -f1  
/flats/F1\_113/history\_113\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 97972 98838 -f1  
/flats/F1\_114/history\_114\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 98839 99705 -f1  
/flats/F1\_115/history\_115\_1.dat

/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 99706 100572 -f1  
/flats/F1\_116/history\_116\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 100573 101439 -f1  
/flats/F1\_117/history\_117\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 101440 102306 -f1  
/flats/F1\_118/history\_118\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 102307 103173 -f1  
/flats/F1\_119/history\_119\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 8 -r 103174 104040 -f1  
/flats/F1\_120/history\_120\_1.dat

## DDL/GEN ITEM ALL.sh

/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 5 -f1 /flats/F1\_001/item\_1.dat

## DDL/GEN NEW ORDER ALL.sh

/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 1 867 -f1  
/flats/F1\_001/neworder\_001\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 868 1734 -f1  
/flats/F1\_002/neworder\_002\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 1735 2601 -f1  
/flats/F1\_003/neworder\_003\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 2602 3468 -f1  
/flats/F1\_004/neworder\_004\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 3469 4335 -f1  
/flats/F1\_005/neworder\_005\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 4336 5202 -f1  
/flats/F1\_006/neworder\_006\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 5203 6069 -f1  
/flats/F1\_007/neworder\_007\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 6070 6936 -f1  
/flats/F1\_008/neworder\_008\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 6937 7803 -f1  
/flats/F1\_009/neworder\_009\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 7804 8670 -f1  
/flats/F1\_010/neworder\_010\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 8671 9537 -f1  
/flats/F1\_011/neworder\_011\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 9538 10404 -f1  
/flats/F1\_012/neworder\_012\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 10405 11271 -f1  
/flats/F1\_013/neworder\_013\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 11272 12138 -f1  
/flats/F1\_014/neworder\_014\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 12139 13005 -f1  
/flats/F1\_015/neworder\_015\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 13006 13872 -f1  
/flats/F1\_016/neworder\_016\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 13873 14739 -f1  
/flats/F1\_017/neworder\_017\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 14740 15606 -f1  
/flats/F1\_018/neworder\_018\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 15607 16473 -f1  
/flats/F1\_019/neworder\_019\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 16474 17340 -f1  
/flats/F1\_020/neworder\_020\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 17341 18207 -f1  
/flats/F1\_021/neworder\_021\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 18208 19074 -f1  
/flats/F1\_022/neworder\_022\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 19075 19941 -f1  
/flats/F1\_023/neworder\_023\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 19942 20808 -f1  
/flats/F1\_024/neworder\_024\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 20809 21675 -f1  
/flats/F1\_025/neworder\_025\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 21676 22542 -f1  
/flats/F1\_026/neworder\_026\_1.dat

/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 22543 23409 -f1  
/flats/F1\_027/neworder\_027\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 23410 24276 -f1  
/flats/F1\_028/neworder\_028\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 24277 25143 -f1  
/flats/F1\_029/neworder\_029\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 25144 26010 -f1  
/flats/F1\_030/neworder\_030\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 26011 26877 -f1  
/flats/F1\_031/neworder\_031\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 26878 27744 -f1  
/flats/F1\_032/neworder\_032\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 27745 28611 -f1  
/flats/F1\_033/neworder\_033\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 28612 29478 -f1  
/flats/F1\_034/neworder\_034\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 29479 30345 -f1  
/flats/F1\_035/neworder\_035\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 30346 31212 -f1  
/flats/F1\_036/neworder\_036\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 31213 32079 -f1  
/flats/F1\_037/neworder\_037\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 32080 32946 -f1  
/flats/F1\_038/neworder\_038\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 32947 33813 -f1  
/flats/F1\_039/neworder\_039\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 33814 34680 -f1  
/flats/F1\_040/neworder\_040\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 34681 35547 -f1  
/flats/F1\_041/neworder\_041\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 35548 36414 -f1  
/flats/F1\_042/neworder\_042\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 36415 37281 -f1  
/flats/F1\_043/neworder\_043\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 37282 38148 -f1  
/flats/F1\_044/neworder\_044\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 38149 39015 -f1  
/flats/F1\_045/neworder\_045\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 39016 39882 -f1  
/flats/F1\_046/neworder\_046\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 39883 40749 -f1  
/flats/F1\_047/neworder\_047\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 40750 41616 -f1  
/flats/F1\_048/neworder\_048\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 41617 42483 -f1  
/flats/F1\_049/neworder\_049\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 42484 43350 -f1  
/flats/F1\_050/neworder\_050\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 43351 44217 -f1  
/flats/F1\_051/neworder\_051\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 44218 45084 -f1  
/flats/F1\_052/neworder\_052\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 45085 45951 -f1  
/flats/F1\_053/neworder\_053\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 45952 46818 -f1  
/flats/F1\_054/neworder\_054\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 46819 47685 -f1  
/flats/F1\_055/neworder\_055\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 47686 48552 -f1  
/flats/F1\_056/neworder\_056\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 48553 49419 -f1  
/flats/F1\_057/neworder\_057\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 49420 50286 -f1  
/flats/F1\_058/neworder\_058\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 50287 51153 -f1  
/flats/F1\_059/neworder\_059\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 51154 52020 -f1  
/flats/F1\_060/neworder\_060\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 52021 52887 -f1  
/flats/F1\_061/neworder\_061\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 11 -r 52888 53754 -f1  
/flats/F1\_062/neworder\_062\_1.dat







/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 19075 19941 -f1  
/flats/F1\_023/warehouse\_023\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 19942 20808 -f1  
/flats/F1\_024/warehouse\_024\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 20809 21675 -f1  
/flats/F1\_025/warehouse\_025\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 21676 22542 -f1  
/flats/F1\_026/warehouse\_026\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 22543 23409 -f1  
/flats/F1\_027/warehouse\_027\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 23410 24276 -f1  
/flats/F1\_028/warehouse\_028\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 24277 25143 -f1  
/flats/F1\_029/warehouse\_029\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 25144 26010 -f1  
/flats/F1\_030/warehouse\_030\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 26011 26877 -f1  
/flats/F1\_031/warehouse\_031\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 26878 27744 -f1  
/flats/F1\_032/warehouse\_032\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 27745 28611 -f1  
/flats/F1\_033/warehouse\_033\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 28612 29478 -f1  
/flats/F1\_034/warehouse\_034\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 29479 30345 -f1  
/flats/F1\_035/warehouse\_035\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 30346 31212 -f1  
/flats/F1\_036/warehouse\_036\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 31213 32079 -f1  
/flats/F1\_037/warehouse\_037\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 32080 32946 -f1  
/flats/F1\_038/warehouse\_038\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 32947 33813 -f1  
/flats/F1\_039/warehouse\_039\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 33814 34680 -f1  
/flats/F1\_040/warehouse\_040\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 34681 35547 -f1  
/flats/F1\_041/warehouse\_041\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 35548 36414 -f1  
/flats/F1\_042/warehouse\_042\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 36415 37281 -f1  
/flats/F1\_043/warehouse\_043\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 37282 38148 -f1  
/flats/F1\_044/warehouse\_044\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 38149 39015 -f1  
/flats/F1\_045/warehouse\_045\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 39016 39882 -f1  
/flats/F1\_046/warehouse\_046\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 39883 40749 -f1  
/flats/F1\_047/warehouse\_047\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 40750 41616 -f1  
/flats/F1\_048/warehouse\_048\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 41617 42483 -f1  
/flats/F1\_049/warehouse\_049\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 42484 43350 -f1  
/flats/F1\_050/warehouse\_050\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 43351 44217 -f1  
/flats/F1\_051/warehouse\_051\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 44218 45084 -f1  
/flats/F1\_052/warehouse\_052\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 45085 45951 -f1  
/flats/F1\_053/warehouse\_053\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 45952 46818 -f1  
/flats/F1\_054/warehouse\_054\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 46819 47685 -f1  
/flats/F1\_055/warehouse\_055\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 47686 48552 -f1  
/flats/F1\_056/warehouse\_056\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 48553 49419 -f1  
/flats/F1\_057/warehouse\_057\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 49420 50286 -f1  
/flats/F1\_058/warehouse\_058\_1.dat

/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 50287 51153 -f1  
/flats/F1\_059/warehouse\_059\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 51154 52020 -f1  
/flats/F1\_060/warehouse\_060\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 52021 52887 -f1  
/flats/F1\_061/warehouse\_061\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 52888 53754 -f1  
/flats/F1\_062/warehouse\_062\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 53755 54621 -f1  
/flats/F1\_063/warehouse\_063\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 54622 55488 -f1  
/flats/F1\_064/warehouse\_064\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 55489 56355 -f1  
/flats/F1\_065/warehouse\_065\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 56356 57222 -f1  
/flats/F1\_066/warehouse\_066\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 57223 58089 -f1  
/flats/F1\_067/warehouse\_067\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 58090 58956 -f1  
/flats/F1\_068/warehouse\_068\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 58957 59823 -f1  
/flats/F1\_069/warehouse\_069\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 59824 60690 -f1  
/flats/F1\_070/warehouse\_070\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 60691 61557 -f1  
/flats/F1\_071/warehouse\_071\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 61558 62424 -f1  
/flats/F1\_072/warehouse\_072\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 62425 63291 -f1  
/flats/F1\_073/warehouse\_073\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 63292 64158 -f1  
/flats/F1\_074/warehouse\_074\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 64159 65025 -f1  
/flats/F1\_075/warehouse\_075\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 65026 65892 -f1  
/flats/F1\_076/warehouse\_076\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 65893 66759 -f1  
/flats/F1\_077/warehouse\_077\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 66760 67626 -f1  
/flats/F1\_078/warehouse\_078\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 67627 68493 -f1  
/flats/F1\_079/warehouse\_079\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 68494 69360 -f1  
/flats/F1\_080/warehouse\_080\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 69361 70227 -f1  
/flats/F1\_081/warehouse\_081\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 70228 71094 -f1  
/flats/F1\_082/warehouse\_082\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 71095 71961 -f1  
/flats/F1\_083/warehouse\_083\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 71962 72828 -f1  
/flats/F1\_084/warehouse\_084\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 72829 73695 -f1  
/flats/F1\_085/warehouse\_085\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 73696 74562 -f1  
/flats/F1\_086/warehouse\_086\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 74563 75429 -f1  
/flats/F1\_087/warehouse\_087\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 75430 76296 -f1  
/flats/F1\_088/warehouse\_088\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 76297 77163 -f1  
/flats/F1\_089/warehouse\_089\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 77164 78030 -f1  
/flats/F1\_090/warehouse\_090\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 78031 78897 -f1  
/flats/F1\_091/warehouse\_091\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 78898 79764 -f1  
/flats/F1\_092/warehouse\_092\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 79765 80631 -f1  
/flats/F1\_093/warehouse\_093\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 80632 81498 -f1  
/flats/F1\_094/warehouse\_094\_1.dat

/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 81499 82365 -f1  
/flats/F1\_095/warehouse\_095\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 82366 83232 -f1  
/flats/F1\_096/warehouse\_096\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 83233 84099 -f1  
/flats/F1\_097/warehouse\_097\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 84100 84966 -f1  
/flats/F1\_098/warehouse\_098\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 84967 85833 -f1  
/flats/F1\_099/warehouse\_099\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 85834 86700 -f1  
/flats/F1\_100/warehouse\_100\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 86701 87567 -f1  
/flats/F1\_101/warehouse\_101\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 87568 88434 -f1  
/flats/F1\_102/warehouse\_102\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 88435 89301 -f1  
/flats/F1\_103/warehouse\_103\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 89302 90168 -f1  
/flats/F1\_104/warehouse\_104\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 90169 91035 -f1  
/flats/F1\_105/warehouse\_105\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 91036 91902 -f1  
/flats/F1\_106/warehouse\_106\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 91903 92769 -f1  
/flats/F1\_107/warehouse\_107\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 92770 93636 -f1  
/flats/F1\_108/warehouse\_108\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 93637 94503 -f1  
/flats/F1\_109/warehouse\_109\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 94504 95370 -f1  
/flats/F1\_110/warehouse\_110\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 95371 96237 -f1  
/flats/F1\_111/warehouse\_111\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 96238 97104 -f1  
/flats/F1\_112/warehouse\_112\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 97105 97971 -f1  
/flats/F1\_113/warehouse\_113\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 97972 98838 -f1  
/flats/F1\_114/warehouse\_114\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 98839 99705 -f1  
/flats/F1\_115/warehouse\_115\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 99706 100572 -f1  
/flats/F1\_116/warehouse\_116\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 100573 101439 -f1  
/flats/F1\_117/warehouse\_117\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 101440 102306 -f1  
/flats/F1\_118/warehouse\_118\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 102307 103173 -f1  
/flats/F1\_119/warehouse\_119\_1.dat  
/home/tpcc/tpc-c.ibm.NEW\_080131\_KIT/dbgen/gendata -t 3 -r 103174 104040 -f1  
/flats/F1\_120/warehouse\_120\_1.dat

## **DDLLOAD CUSTOMER ALL.ddl**

```
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;  
IMPORT FROM /flats/F1_001/customer_001_1.dat OF DEL MODIFIED BY COLDEL  
TIMESTAMPFORMAT='YYYY-MM-DD HH:MM:SS' KEEPBLANKS COMPOUND=50 COMMITCOUNT  
26010000 INSERT INTO CUSTOMER1;  
COMMIT WORK;  
CONNECT RESET;  
CONNECT TO TPCC IN SHARE MODE;  
UPDATE COMMAND OPTIONS USING C OFF;  
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;  
IMPORT FROM /flats/F1_002/customer_002_1.dat OF DEL MODIFIED BY COLDEL  
TIMESTAMPFORMAT='YYYY-MM-DD HH:MM:SS' KEEPBLANKS COMPOUND=50 COMMITCOUNT  
26010000 INSERT INTO CUSTOMER2;  
COMMIT WORK;  
CONNECT RESET;
```



































































































































```
ALTER TABLESPACE ts_newordB_117 BUFFERPOOL IBMDEFAULTBP;
ALTER TABLESPACE ts_newordB_118 BUFFERPOOL IBMDEFAULTBP;
ALTER TABLESPACE ts_newordB_119 BUFFERPOOL IBMDEFAULTBP;
ALTER TABLESPACE ts_newordB_120 BUFFERPOOL IBMDEFAULTBP;
```

## bp/create bufferpool split.ddl

```
CREATE BUFFERPOOL WAR1 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL WAR2 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL WAR3 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL WAR4 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL WAR5 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL WAR6 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL WAR7 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL WAR8 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS1 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS2 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS3 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS4 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS5 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS6 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS7 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL DIS8 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL ITM SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL HST1 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL HST2 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL HST3 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL HST4 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL HST5 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL HST6 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL HST7 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL HST8 SIZE 100 PAGESIZE 16384;
CREATE BUFFERPOOL NEW1 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL NEW2 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL NEW3 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL NEW4 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL NEW5 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL NEW6 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL NEW7 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL NEW8 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL ORD1 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD2 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD3 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD4 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD5 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD6 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD7 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD8 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I1 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I2 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I3 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I4 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I5 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I6 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I7 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL ORD_I8 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN1 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN2 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN3 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN4 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN5 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN6 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN7 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL OLN8 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST1 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL CST2 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL CST3 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL CST4 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL CST5 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL CST6 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL CST7 SIZE 100 PAGESIZE 4096;
```

```
CREATE BUFFERPOOL CST8 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL CST_I1 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST_I2 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST_I3 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST_I4 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST_I5 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST_I6 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST_I7 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL CST_I8 SIZE 100 PAGESIZE 8192;
CREATE BUFFERPOOL STK1 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL STK2 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL STK3 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL STK4 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL STK5 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL STK6 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL STK7 SIZE 100 PAGESIZE 4096;
CREATE BUFFERPOOL STK8 SIZE 100 PAGESIZE 4096;
```

## bp/create default bufferpool.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 - 2005
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----
-- create_bufferpool.ddl.sample - Sample Create Bufferpool DDL
--
```

```
connect to tpcc;
create bufferpool IBMDEFAULTBP8K size 1000 pagesize 8192;
create bufferpool IBMDEFAULTBP16K size 1000 pagesize 16384;
connect reset;
terminate;
```

## db/create database.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 - 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-----
```

```
drop database tpcc;
create database tpcc on /home/tpcc/db/tpccdb1 collate using identity
catalog tablespace
managed by system using ('/home/tpcc/db/db1catalog');
```

## ts/cris customer.ddl

```
connect to tpcc;
-- now creating TS for is_customer_001 of D1
```

```
drop tablespace is_customer_001;
create regular tablespace is_customer_001 pagesize 8K
```

```
managed by database
using
(
    device '/dev/rD1F01V1CST1' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_002 of D1

drop tablespace is_customer_002;
create regular tablespace is_customer_002 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2CST1' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_003 of D1

drop tablespace is_customer_003;
create regular tablespace is_customer_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3CST1' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_004 of D1

drop tablespace is_customer_004;
create regular tablespace is_customer_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4CST1' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_005 of D1

drop tablespace is_customer_005;
create regular tablespace is_customer_005 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V5CST1' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_006 of D1

drop tablespace is_customer_006;
create regular tablespace is_customer_006 pagesize 8K
managed by database
using
(
```

```

        device '/dev/rD1F01V6CSTI' 188928
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_007 of D1

drop tablespace is_customer_007;
create regular tablespace is_customer_007 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V7CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_008 of D1

drop tablespace is_customer_008;
create regular tablespace is_customer_008 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V8CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_009 of D1

drop tablespace is_customer_009;
create regular tablespace is_customer_009 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V9CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_010 of D1

drop tablespace is_customer_010;
create regular tablespace is_customer_010 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V10CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_011 of D1

drop tablespace is_customer_011;
create regular tablespace is_customer_011 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V11CSTI' 188928
)
extentsize 256

```

```

    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_012 of D1

drop tablespace is_customer_012;
create regular tablespace is_customer_012 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V12CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_013 of D1

drop tablespace is_customer_013;
create regular tablespace is_customer_013 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V13CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_014 of D1

drop tablespace is_customer_014;
create regular tablespace is_customer_014 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V14CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_015 of D1

drop tablespace is_customer_015;
create regular tablespace is_customer_015 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V15CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_016 of D1

drop tablespace is_customer_016;
create regular tablespace is_customer_016 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V1CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_017 of D1

drop tablespace is_customer_017;
create regular tablespace is_customer_017 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V2CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_018 of D1

drop tablespace is_customer_018;
create regular tablespace is_customer_018 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V3CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_019 of D1

drop tablespace is_customer_019;
create regular tablespace is_customer_019 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V4CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_020 of D1

drop tablespace is_customer_020;
create regular tablespace is_customer_020 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V5CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_021 of D1

drop tablespace is_customer_021;
create regular tablespace is_customer_021 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V6CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_022 of D1

```

```

drop tablespace is_customer_022;
create regular tablespace is_customer_022 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V7CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_023 of D1

drop tablespace is_customer_023;
create regular tablespace is_customer_023 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V8CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_024 of D1

drop tablespace is_customer_024;
create regular tablespace is_customer_024 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V9CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_025 of D1

drop tablespace is_customer_025;
create regular tablespace is_customer_025 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V10CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_026 of D1

drop tablespace is_customer_026;
create regular tablespace is_customer_026 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V11CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_027 of D1

drop tablespace is_customer_027;
create regular tablespace is_customer_027 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F02V12CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_028 of D1

drop tablespace is_customer_028;
create regular tablespace is_customer_028 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V13CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_029 of D1

drop tablespace is_customer_029;
create regular tablespace is_customer_029 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V14CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_030 of D1

drop tablespace is_customer_030;
create regular tablespace is_customer_030 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V15CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_031 of D1

drop tablespace is_customer_031;
create regular tablespace is_customer_031 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V1CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_032 of D1

drop tablespace is_customer_032;
create regular tablespace is_customer_032 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V2CSTI' 188928

```

```

)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_033 of D1

drop tablespace is_customer_033;
create regular tablespace is_customer_033 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V3CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_034 of D1

drop tablespace is_customer_034;
create regular tablespace is_customer_034 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V4CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_035 of D1

drop tablespace is_customer_035;
create regular tablespace is_customer_035 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V5CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_036 of D1

drop tablespace is_customer_036;
create regular tablespace is_customer_036 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V6CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_037 of D1

drop tablespace is_customer_037;
create regular tablespace is_customer_037 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V7CSTI' 188928
)
extentsize 256
prefetchsize 4096

```

```

        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_038 of D1
drop tablespace is_customer_038;
create regular tablespace is_customer_038 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V8CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_039 of D1
drop tablespace is_customer_039;
create regular tablespace is_customer_039 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V9CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_040 of D1
drop tablespace is_customer_040;
create regular tablespace is_customer_040 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V10CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_041 of D1
drop tablespace is_customer_041;
create regular tablespace is_customer_041 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V11CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_042 of D1
drop tablespace is_customer_042;
create regular tablespace is_customer_042 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V12CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_043 of D1
drop tablespace is_customer_043;
create regular tablespace is_customer_043 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V13CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_044 of D1
drop tablespace is_customer_044;
create regular tablespace is_customer_044 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V14CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_045 of D1
drop tablespace is_customer_045;
create regular tablespace is_customer_045 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V15CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_046 of D1
drop tablespace is_customer_046;
create regular tablespace is_customer_046 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V1CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_047 of D1
drop tablespace is_customer_047;
create regular tablespace is_customer_047 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V2CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_048 of D1
drop tablespace is_customer_048;

```

```

create regular tablespace is_customer_048 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V3CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_049 of D1
drop tablespace is_customer_049;
create regular tablespace is_customer_049 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V4CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_050 of D1
drop tablespace is_customer_050;
create regular tablespace is_customer_050 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V5CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_051 of D1
drop tablespace is_customer_051;
create regular tablespace is_customer_051 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V6CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_052 of D1
drop tablespace is_customer_052;
create regular tablespace is_customer_052 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V7CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_053 of D1
drop tablespace is_customer_053;
create regular tablespace is_customer_053 pagesize 8K
managed by database
using

```

```

(
  device '/dev/rD1F04V8CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_054 of D1

drop tablespace is_customer_054;
create regular tablespace is_customer_054 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V9CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_055 of D1

drop tablespace is_customer_055;
create regular tablespace is_customer_055 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V10CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_056 of D1

drop tablespace is_customer_056;
create regular tablespace is_customer_056 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V11CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_057 of D1

drop tablespace is_customer_057;
create regular tablespace is_customer_057 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V12CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_058 of D1

drop tablespace is_customer_058;
create regular tablespace is_customer_058 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V13CSTI' 188928
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_059 of D1

drop tablespace is_customer_059;
create regular tablespace is_customer_059 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V14CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_060 of D1

drop tablespace is_customer_060;
create regular tablespace is_customer_060 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V15CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_061 of D1

drop tablespace is_customer_061;
create regular tablespace is_customer_061 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V1CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_062 of D1

drop tablespace is_customer_062;
create regular tablespace is_customer_062 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V2CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_063 of D1

drop tablespace is_customer_063;
create regular tablespace is_customer_063 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V3CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for is_customer_064 of D1

drop tablespace is_customer_064;
create regular tablespace is_customer_064 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V4CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_065 of D1

drop tablespace is_customer_065;
create regular tablespace is_customer_065 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V5CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_066 of D1

drop tablespace is_customer_066;
create regular tablespace is_customer_066 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V6CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_067 of D1

drop tablespace is_customer_067;
create regular tablespace is_customer_067 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V7CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_068 of D1

drop tablespace is_customer_068;
create regular tablespace is_customer_068 pagesize 8K
managed by database
using
(
  device '/dev/rD1F05V8CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_069 of D1

```

```

drop tablespace is_customer_069;
create regular tablespace is_customer_069 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V9CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_070 of D1

drop tablespace is_customer_070;
create regular tablespace is_customer_070 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V10CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_071 of D1

drop tablespace is_customer_071;
create regular tablespace is_customer_071 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V11CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_072 of D1

drop tablespace is_customer_072;
create regular tablespace is_customer_072 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V12CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_073 of D1

drop tablespace is_customer_073;
create regular tablespace is_customer_073 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V13CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_074 of D1

drop tablespace is_customer_074;
create regular tablespace is_customer_074 pagesize 8K

```

```

  managed by database
  using
  (
    device '/dev/rD1F05V14CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_075 of D1

drop tablespace is_customer_075;
create regular tablespace is_customer_075 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V15CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_076 of D1

drop tablespace is_customer_076;
create regular tablespace is_customer_076 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V1CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_077 of D1

drop tablespace is_customer_077;
create regular tablespace is_customer_077 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V2CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_078 of D1

drop tablespace is_customer_078;
create regular tablespace is_customer_078 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V3CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_079 of D1

drop tablespace is_customer_079;
create regular tablespace is_customer_079 pagesize 8K
  managed by database
  using
  (

```

```

    device '/dev/rD1F06V4CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_080 of D1

drop tablespace is_customer_080;
create regular tablespace is_customer_080 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V5CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_081 of D1

drop tablespace is_customer_081;
create regular tablespace is_customer_081 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V6CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_082 of D1

drop tablespace is_customer_082;
create regular tablespace is_customer_082 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V7CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_083 of D1

drop tablespace is_customer_083;
create regular tablespace is_customer_083 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V8CSTI' 188928
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_084 of D1

drop tablespace is_customer_084;
create regular tablespace is_customer_084 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V9CSTI' 188928
  )
  extentsize 256

```

```

        prefetchsize 4096
        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_085 of D1
drop tablespace is_customer_085;
create regular tablespace is_customer_085 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V10CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_086 of D1
drop tablespace is_customer_086;
create regular tablespace is_customer_086 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V11CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_087 of D1
drop tablespace is_customer_087;
create regular tablespace is_customer_087 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V12CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_088 of D1
drop tablespace is_customer_088;
create regular tablespace is_customer_088 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V13CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_089 of D1
drop tablespace is_customer_089;
create regular tablespace is_customer_089 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V14CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_090 of D1
drop tablespace is_customer_090;
create regular tablespace is_customer_090 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V15CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_091 of D1
drop tablespace is_customer_091;
create regular tablespace is_customer_091 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V1CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_092 of D1
drop tablespace is_customer_092;
create regular tablespace is_customer_092 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V2CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_093 of D1
drop tablespace is_customer_093;
create regular tablespace is_customer_093 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V3CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_094 of D1
drop tablespace is_customer_094;
create regular tablespace is_customer_094 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V4CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_095 of D1

```

```

drop tablespace is_customer_095;
create regular tablespace is_customer_095 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V5CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_096 of D1
drop tablespace is_customer_096;
create regular tablespace is_customer_096 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V6CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_097 of D1
drop tablespace is_customer_097;
create regular tablespace is_customer_097 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V7CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_098 of D1
drop tablespace is_customer_098;
create regular tablespace is_customer_098 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V8CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_099 of D1
drop tablespace is_customer_099;
create regular tablespace is_customer_099 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V9CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_100 of D1
drop tablespace is_customer_100;
create regular tablespace is_customer_100 pagesize 8K
managed by database

```



```

using
(
    device '/dev/rD1F07V10CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_101 of D1

drop tablespace is_customer_101;
create regular tablespace is_customer_101 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V11CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_102 of D1

drop tablespace is_customer_102;
create regular tablespace is_customer_102 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V12CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_103 of D1

drop tablespace is_customer_103;
create regular tablespace is_customer_103 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V13CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_104 of D1

drop tablespace is_customer_104;
create regular tablespace is_customer_104 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V14CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_105 of D1

drop tablespace is_customer_105;
create regular tablespace is_customer_105 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V15CSTI' 188928

```

```

)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_106 of D1

drop tablespace is_customer_106;
create regular tablespace is_customer_106 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V1CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_107 of D1

drop tablespace is_customer_107;
create regular tablespace is_customer_107 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V2CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_108 of D1

drop tablespace is_customer_108;
create regular tablespace is_customer_108 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V3CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_109 of D1

drop tablespace is_customer_109;
create regular tablespace is_customer_109 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V4CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_110 of D1

drop tablespace is_customer_110;
create regular tablespace is_customer_110 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V5CSTI' 188928
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_111 of D1

drop tablespace is_customer_111;
create regular tablespace is_customer_111 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V6CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_112 of D1

drop tablespace is_customer_112;
create regular tablespace is_customer_112 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V7CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_113 of D1

drop tablespace is_customer_113;
create regular tablespace is_customer_113 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V8CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_114 of D1

drop tablespace is_customer_114;
create regular tablespace is_customer_114 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V9CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_115 of D1

drop tablespace is_customer_115;
create regular tablespace is_customer_115 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V10CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_116 of D1

drop tablespace is_customer_116;
create regular tablespace is_customer_116 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V11CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_117 of D1

drop tablespace is_customer_117;
create regular tablespace is_customer_117 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V12CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_118 of D1

drop tablespace is_customer_118;
create regular tablespace is_customer_118 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V13CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_119 of D1

drop tablespace is_customer_119;
create regular tablespace is_customer_119 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V14CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_120 of D1

drop tablespace is_customer_120;
create regular tablespace is_customer_120 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V15CSTI' 188928
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

### ts/cris\_order.ddl

```

connect to tpcc;
-- now creating TS for is_order_001 of D1

drop tablespace is_order_001;
create regular tablespace is_order_001 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_002 of D1

drop tablespace is_order_002;
create regular tablespace is_order_002 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_003 of D1

drop tablespace is_order_003;
create regular tablespace is_order_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_004 of D1

drop tablespace is_order_004;
create regular tablespace is_order_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_005 of D1

drop tablespace is_order_005;
create regular tablespace is_order_005 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V5ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_006 of D1

drop tablespace is_order_006;
create regular tablespace is_order_006 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V6ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_007 of D1

drop tablespace is_order_007;
create regular tablespace is_order_007 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V7ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_008 of D1

drop tablespace is_order_008;
create regular tablespace is_order_008 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V8ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_009 of D1

drop tablespace is_order_009;
create regular tablespace is_order_009 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V9ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_010 of D1

drop tablespace is_order_010;
create regular tablespace is_order_010 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V10ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_011 of D1

```

```

drop tablespace is_order_011;
create regular tablespace is_order_011 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V11ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_012 of D1

drop tablespace is_order_012;
create regular tablespace is_order_012 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V12ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_013 of D1

drop tablespace is_order_013;
create regular tablespace is_order_013 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V13ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_014 of D1

drop tablespace is_order_014;
create regular tablespace is_order_014 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V14ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_015 of D1

drop tablespace is_order_015;
create regular tablespace is_order_015 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V15ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_016 of D1

drop tablespace is_order_016;
create regular tablespace is_order_016 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F02V1ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_017 of D1

drop tablespace is_order_017;
create regular tablespace is_order_017 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V2ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_018 of D1

drop tablespace is_order_018;
create regular tablespace is_order_018 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V3ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_019 of D1

drop tablespace is_order_019;
create regular tablespace is_order_019 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V4ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_020 of D1

drop tablespace is_order_020;
create regular tablespace is_order_020 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V5ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_021 of D1

drop tablespace is_order_021;
create regular tablespace is_order_021 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V6ORDI' 125952

```

```

)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_022 of D1

drop tablespace is_order_022;
create regular tablespace is_order_022 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V7ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_023 of D1

drop tablespace is_order_023;
create regular tablespace is_order_023 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V8ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_024 of D1

drop tablespace is_order_024;
create regular tablespace is_order_024 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V9ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_025 of D1

drop tablespace is_order_025;
create regular tablespace is_order_025 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V10ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_026 of D1

drop tablespace is_order_026;
create regular tablespace is_order_026 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V11ORDI' 125952
)
extentsize 256
prefetchsize 4096

```

```

        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_027 of D1
drop tablespace is_order_027;
create regular tablespace is_order_027 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V12ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_028 of D1
drop tablespace is_order_028;
create regular tablespace is_order_028 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V13ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_029 of D1
drop tablespace is_order_029;
create regular tablespace is_order_029 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V14ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_030 of D1
drop tablespace is_order_030;
create regular tablespace is_order_030 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V15ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_031 of D1
drop tablespace is_order_031;
create regular tablespace is_order_031 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V1ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_032 of D1
drop tablespace is_order_032;
create regular tablespace is_order_032 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V2ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_033 of D1
drop tablespace is_order_033;
create regular tablespace is_order_033 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V3ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_034 of D1
drop tablespace is_order_034;
create regular tablespace is_order_034 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V4ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_035 of D1
drop tablespace is_order_035;
create regular tablespace is_order_035 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V5ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_036 of D1
drop tablespace is_order_036;
create regular tablespace is_order_036 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V6ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_037 of D1
drop tablespace is_order_037;

```

```

create regular tablespace is_order_037 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V7ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_038 of D1
drop tablespace is_order_038;
create regular tablespace is_order_038 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V8ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_039 of D1
drop tablespace is_order_039;
create regular tablespace is_order_039 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V9ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_040 of D1
drop tablespace is_order_040;
create regular tablespace is_order_040 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V10ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_041 of D1
drop tablespace is_order_041;
create regular tablespace is_order_041 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V11ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_042 of D1
drop tablespace is_order_042;
create regular tablespace is_order_042 pagesize 8K
managed by database
using

```

```

(
  device '/dev/rD1F03V12ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_043 of D1

drop tablespace is_order_043;
create regular tablespace is_order_043 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V13ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_044 of D1

drop tablespace is_order_044;
create regular tablespace is_order_044 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V14ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_045 of D1

drop tablespace is_order_045;
create regular tablespace is_order_045 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V15ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_046 of D1

drop tablespace is_order_046;
create regular tablespace is_order_046 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V1ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_047 of D1

drop tablespace is_order_047;
create regular tablespace is_order_047 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V2ORDI' 125952
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_048 of D1

drop tablespace is_order_048;
create regular tablespace is_order_048 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V3ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_049 of D1

drop tablespace is_order_049;
create regular tablespace is_order_049 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V4ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_050 of D1

drop tablespace is_order_050;
create regular tablespace is_order_050 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V5ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_051 of D1

drop tablespace is_order_051;
create regular tablespace is_order_051 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V6ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_052 of D1

drop tablespace is_order_052;
create regular tablespace is_order_052 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V7ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for is_order_053 of D1

drop tablespace is_order_053;
create regular tablespace is_order_053 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V8ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_054 of D1

drop tablespace is_order_054;
create regular tablespace is_order_054 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V9ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_055 of D1

drop tablespace is_order_055;
create regular tablespace is_order_055 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V10ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_056 of D1

drop tablespace is_order_056;
create regular tablespace is_order_056 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V11ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_057 of D1

drop tablespace is_order_057;
create regular tablespace is_order_057 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V12ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_058 of D1

```

```

drop tablespace is_order_058;
create regular tablespace is_order_058 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V13ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_059 of D1

drop tablespace is_order_059;
create regular tablespace is_order_059 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V14ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_060 of D1

drop tablespace is_order_060;
create regular tablespace is_order_060 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V15ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_061 of D1

drop tablespace is_order_061;
create regular tablespace is_order_061 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V1ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_062 of D1

drop tablespace is_order_062;
create regular tablespace is_order_062 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V2ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_063 of D1

drop tablespace is_order_063;
create regular tablespace is_order_063 pagesize 8K

```

```

  managed by database
  using
  (
    device '/dev/rD1F05V3ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_064 of D1

drop tablespace is_order_064;
create regular tablespace is_order_064 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V4ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_065 of D1

drop tablespace is_order_065;
create regular tablespace is_order_065 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V5ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_066 of D1

drop tablespace is_order_066;
create regular tablespace is_order_066 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V6ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_067 of D1

drop tablespace is_order_067;
create regular tablespace is_order_067 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V7ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_068 of D1

drop tablespace is_order_068;
create regular tablespace is_order_068 pagesize 8K
  managed by database
  using
  (

```

```

    device '/dev/rD1F05V8ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_069 of D1

drop tablespace is_order_069;
create regular tablespace is_order_069 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V9ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_070 of D1

drop tablespace is_order_070;
create regular tablespace is_order_070 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V10ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_071 of D1

drop tablespace is_order_071;
create regular tablespace is_order_071 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V11ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_072 of D1

drop tablespace is_order_072;
create regular tablespace is_order_072 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V12ORDI' 125952
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_073 of D1

drop tablespace is_order_073;
create regular tablespace is_order_073 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V13ORDI' 125952
  )
  extentsize 256

```

```

        prefetchsize 4096
        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_074 of D1
drop tablespace is_order_074;
create regular tablespace is_order_074 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V14ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_075 of D1
drop tablespace is_order_075;
create regular tablespace is_order_075 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V15ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_076 of D1
drop tablespace is_order_076;
create regular tablespace is_order_076 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V1ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_077 of D1
drop tablespace is_order_077;
create regular tablespace is_order_077 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V2ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_078 of D1
drop tablespace is_order_078;
create regular tablespace is_order_078 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V3ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_079 of D1
drop tablespace is_order_079;
create regular tablespace is_order_079 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V4ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_080 of D1
drop tablespace is_order_080;
create regular tablespace is_order_080 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V5ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_081 of D1
drop tablespace is_order_081;
create regular tablespace is_order_081 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V6ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_082 of D1
drop tablespace is_order_082;
create regular tablespace is_order_082 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V7ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_083 of D1
drop tablespace is_order_083;
create regular tablespace is_order_083 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V8ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_084 of D1

```

```

drop tablespace is_order_084;
create regular tablespace is_order_084 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V9ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_085 of D1
drop tablespace is_order_085;
create regular tablespace is_order_085 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V10ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_086 of D1
drop tablespace is_order_086;
create regular tablespace is_order_086 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V11ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_087 of D1
drop tablespace is_order_087;
create regular tablespace is_order_087 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V12ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_088 of D1
drop tablespace is_order_088;
create regular tablespace is_order_088 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V13ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_089 of D1
drop tablespace is_order_089;
create regular tablespace is_order_089 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F06V14ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_090 of D1
drop tablespace is_order_090;
create regular tablespace is_order_090 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V15ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_091 of D1
drop tablespace is_order_091;
create regular tablespace is_order_091 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V1ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_092 of D1
drop tablespace is_order_092;
create regular tablespace is_order_092 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V2ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_093 of D1
drop tablespace is_order_093;
create regular tablespace is_order_093 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V3ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_094 of D1
drop tablespace is_order_094;
create regular tablespace is_order_094 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V4ORDI' 125952

```

```

)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_095 of D1
drop tablespace is_order_095;
create regular tablespace is_order_095 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V5ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_096 of D1
drop tablespace is_order_096;
create regular tablespace is_order_096 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V6ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_097 of D1
drop tablespace is_order_097;
create regular tablespace is_order_097 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V7ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_098 of D1
drop tablespace is_order_098;
create regular tablespace is_order_098 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V8ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_099 of D1
drop tablespace is_order_099;
create regular tablespace is_order_099 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V9ORDI' 125952
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_100 of D1
drop tablespace is_order_100;
create regular tablespace is_order_100 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V10ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_101 of D1
drop tablespace is_order_101;
create regular tablespace is_order_101 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V11ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_102 of D1
drop tablespace is_order_102;
create regular tablespace is_order_102 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V12ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_103 of D1
drop tablespace is_order_103;
create regular tablespace is_order_103 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V13ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_104 of D1
drop tablespace is_order_104;
create regular tablespace is_order_104 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V14ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```



```

-- now creating TS for is_order_105 of D1
drop tablespace is_order_105;
create regular tablespace is_order_105 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V15ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_106 of D1
drop tablespace is_order_106;
create regular tablespace is_order_106 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V1ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_107 of D1
drop tablespace is_order_107;
create regular tablespace is_order_107 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V2ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_108 of D1
drop tablespace is_order_108;
create regular tablespace is_order_108 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V3ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_109 of D1
drop tablespace is_order_109;
create regular tablespace is_order_109 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V4ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_110 of D1
drop tablespace is_order_110;

```

```

create regular tablespace is_order_110 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V5ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_111 of D1
drop tablespace is_order_111;
create regular tablespace is_order_111 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V6ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_112 of D1
drop tablespace is_order_112;
create regular tablespace is_order_112 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V7ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_113 of D1
drop tablespace is_order_113;
create regular tablespace is_order_113 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V8ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_114 of D1
drop tablespace is_order_114;
create regular tablespace is_order_114 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V9ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_115 of D1
drop tablespace is_order_115;
create regular tablespace is_order_115 pagesize 8K
managed by database
using

```

```

(
    device '/dev/rD1F08V10ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_116 of D1
drop tablespace is_order_116;
create regular tablespace is_order_116 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V11ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_117 of D1
drop tablespace is_order_117;
create regular tablespace is_order_117 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V12ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_118 of D1
drop tablespace is_order_118;
create regular tablespace is_order_118 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V13ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_119 of D1
drop tablespace is_order_119;
create regular tablespace is_order_119 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V14ORDI' 125952
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_120 of D1
drop tablespace is_order_120;
create regular tablespace is_order_120 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V15ORDI' 125952
)

```

```

        extentsize 256
        prefetchsize 4096
        bufferpool ibmdefaultbp8K;
commit;

connect reset;

ts/crts_customer.ddl

connect to tpcc;
-- now creating TS for ts_customer_001 of D1

drop tablespace ts_customer_001;
create regular tablespace ts_customer_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_002 of D1

drop tablespace ts_customer_002;
create regular tablespace ts_customer_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_003 of D1

drop tablespace ts_customer_003;
create regular tablespace ts_customer_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_004 of D1

drop tablespace ts_customer_004;
create regular tablespace ts_customer_004 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V4CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_005 of D1

drop tablespace ts_customer_005;
create regular tablespace ts_customer_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5CST' 5463808

```

```

    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_customer_006 of D1

drop tablespace ts_customer_006;
create regular tablespace ts_customer_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V6CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_007 of D1

drop tablespace ts_customer_007;
create regular tablespace ts_customer_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V7CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_008 of D1

drop tablespace ts_customer_008;
create regular tablespace ts_customer_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V8CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_009 of D1

drop tablespace ts_customer_009;
create regular tablespace ts_customer_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V9CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_010 of D1

drop tablespace ts_customer_010;
create regular tablespace ts_customer_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V10CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_011 of D1

drop tablespace ts_customer_011;

```

```

create regular tablespace ts_customer_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V11CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_012 of D1

drop tablespace ts_customer_012;
create regular tablespace ts_customer_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V12CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_013 of D1

drop tablespace ts_customer_013;
create regular tablespace ts_customer_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V13CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_014 of D1

drop tablespace ts_customer_014;
create regular tablespace ts_customer_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V14CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_015 of D1

drop tablespace ts_customer_015;
create regular tablespace ts_customer_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V15CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_016 of D1

drop tablespace ts_customer_016;
create regular tablespace ts_customer_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1CST' 5463808
)
extentsize 256
prefetchsize 4096;

```

```

commit;
-- now creating TS for ts_customer_017 of D1
drop tablespace ts_customer_017;
create regular tablespace ts_customer_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_018 of D1
drop tablespace ts_customer_018;
create regular tablespace ts_customer_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_019 of D1
drop tablespace ts_customer_019;
create regular tablespace ts_customer_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_020 of D1
drop tablespace ts_customer_020;
create regular tablespace ts_customer_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_021 of D1
drop tablespace ts_customer_021;
create regular tablespace ts_customer_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V6CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_022 of D1
drop tablespace ts_customer_022;
create regular tablespace ts_customer_022 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F02V7CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_023 of D1
drop tablespace ts_customer_023;
create regular tablespace ts_customer_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V8CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_024 of D1
drop tablespace ts_customer_024;
create regular tablespace ts_customer_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V9CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_025 of D1
drop tablespace ts_customer_025;
create regular tablespace ts_customer_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V10CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_026 of D1
drop tablespace ts_customer_026;
create regular tablespace ts_customer_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V11CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_027 of D1
drop tablespace ts_customer_027;
create regular tablespace ts_customer_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V12CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_028 of D1

```

```

drop tablespace ts_customer_028;
create regular tablespace ts_customer_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V13CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_029 of D1
drop tablespace ts_customer_029;
create regular tablespace ts_customer_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V14CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_030 of D1
drop tablespace ts_customer_030;
create regular tablespace ts_customer_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V15CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_031 of D1
drop tablespace ts_customer_031;
create regular tablespace ts_customer_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_032 of D1
drop tablespace ts_customer_032;
create regular tablespace ts_customer_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_033 of D1
drop tablespace ts_customer_033;
create regular tablespace ts_customer_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3CST' 5463808
)

```

```

        extentsize 256
        prefetchsize 4096;
commit;
-- now creating TS for ts_customer_034 of D1
drop tablespace ts_customer_034;
create regular tablespace ts_customer_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_035 of D1
drop tablespace ts_customer_035;
create regular tablespace ts_customer_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_036 of D1
drop tablespace ts_customer_036;
create regular tablespace ts_customer_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V6CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_037 of D1
drop tablespace ts_customer_037;
create regular tablespace ts_customer_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V7CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_038 of D1
drop tablespace ts_customer_038;
create regular tablespace ts_customer_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V8CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_039 of D1
drop tablespace ts_customer_039;
create regular tablespace ts_customer_039 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F03V9CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_040 of D1
drop tablespace ts_customer_040;
create regular tablespace ts_customer_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V10CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_041 of D1
drop tablespace ts_customer_041;
create regular tablespace ts_customer_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V11CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_042 of D1
drop tablespace ts_customer_042;
create regular tablespace ts_customer_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V12CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_043 of D1
drop tablespace ts_customer_043;
create regular tablespace ts_customer_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V13CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_044 of D1
drop tablespace ts_customer_044;
create regular tablespace ts_customer_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V14CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_customer_045 of D1
drop tablespace ts_customer_045;
create regular tablespace ts_customer_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V15CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_046 of D1
drop tablespace ts_customer_046;
create regular tablespace ts_customer_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_047 of D1
drop tablespace ts_customer_047;
create regular tablespace ts_customer_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_048 of D1
drop tablespace ts_customer_048;
create regular tablespace ts_customer_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_049 of D1
drop tablespace ts_customer_049;
create regular tablespace ts_customer_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_050 of D1
drop tablespace ts_customer_050;
create regular tablespace ts_customer_050 pagesize 4K
managed by database
using
(

```

```

        device '/dev/rD1F04V5CST' 5463808
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_customer_051 of D1

drop tablespace ts_customer_051;
create regular tablespace ts_customer_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V6CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_052 of D1

drop tablespace ts_customer_052;
create regular tablespace ts_customer_052 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V7CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_053 of D1

drop tablespace ts_customer_053;
create regular tablespace ts_customer_053 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V8CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_054 of D1

drop tablespace ts_customer_054;
create regular tablespace ts_customer_054 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V9CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_055 of D1

drop tablespace ts_customer_055;
create regular tablespace ts_customer_055 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V10CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_056 of D1

```

```

drop tablespace ts_customer_056;
create regular tablespace ts_customer_056 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V11CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_057 of D1

drop tablespace ts_customer_057;
create regular tablespace ts_customer_057 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V12CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_058 of D1

drop tablespace ts_customer_058;
create regular tablespace ts_customer_058 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V13CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_059 of D1

drop tablespace ts_customer_059;
create regular tablespace ts_customer_059 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V14CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_060 of D1

drop tablespace ts_customer_060;
create regular tablespace ts_customer_060 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V15CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_061 of D1

drop tablespace ts_customer_061;
create regular tablespace ts_customer_061 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1CST' 5463808
)
extentsize 256

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_customer_062 of D1

drop tablespace ts_customer_062;
create regular tablespace ts_customer_062 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_063 of D1

drop tablespace ts_customer_063;
create regular tablespace ts_customer_063 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_064 of D1

drop tablespace ts_customer_064;
create regular tablespace ts_customer_064 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_065 of D1

drop tablespace ts_customer_065;
create regular tablespace ts_customer_065 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_066 of D1

drop tablespace ts_customer_066;
create regular tablespace ts_customer_066 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V6CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_067 of D1

drop tablespace ts_customer_067;
create regular tablespace ts_customer_067 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F05V7CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_068 of D1

drop tablespace ts_customer_068;
create regular tablespace ts_customer_068 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V8CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_069 of D1

drop tablespace ts_customer_069;
create regular tablespace ts_customer_069 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V9CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_070 of D1

drop tablespace ts_customer_070;
create regular tablespace ts_customer_070 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V10CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_071 of D1

drop tablespace ts_customer_071;
create regular tablespace ts_customer_071 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V11CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_072 of D1

drop tablespace ts_customer_072;
create regular tablespace ts_customer_072 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V12CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_customer_073 of D1

drop tablespace ts_customer_073;
create regular tablespace ts_customer_073 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V13CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_074 of D1

drop tablespace ts_customer_074;
create regular tablespace ts_customer_074 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V14CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_075 of D1

drop tablespace ts_customer_075;
create regular tablespace ts_customer_075 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V15CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_076 of D1

drop tablespace ts_customer_076;
create regular tablespace ts_customer_076 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_077 of D1

drop tablespace ts_customer_077;
create regular tablespace ts_customer_077 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_078 of D1

drop tablespace ts_customer_078;
create regular tablespace ts_customer_078 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3CST' 5463808

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_079 of D1

drop tablespace ts_customer_079;
create regular tablespace ts_customer_079 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_080 of D1

drop tablespace ts_customer_080;
create regular tablespace ts_customer_080 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_081 of D1

drop tablespace ts_customer_081;
create regular tablespace ts_customer_081 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V6CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_082 of D1

drop tablespace ts_customer_082;
create regular tablespace ts_customer_082 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V7CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_083 of D1

drop tablespace ts_customer_083;
create regular tablespace ts_customer_083 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V8CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_084 of D1

drop tablespace ts_customer_084;

```

```

create regular tablespace ts_customer_084 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V9CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_085 of D1

drop tablespace ts_customer_085;
create regular tablespace ts_customer_085 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V10CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_086 of D1

drop tablespace ts_customer_086;
create regular tablespace ts_customer_086 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V11CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_087 of D1

drop tablespace ts_customer_087;
create regular tablespace ts_customer_087 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V12CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_088 of D1

drop tablespace ts_customer_088;
create regular tablespace ts_customer_088 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V13CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_089 of D1

drop tablespace ts_customer_089;
create regular tablespace ts_customer_089 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V14CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_customer_090 of D1

drop tablespace ts_customer_090;
create regular tablespace ts_customer_090 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V15CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_091 of D1

drop tablespace ts_customer_091;
create regular tablespace ts_customer_091 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V1CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_092 of D1

drop tablespace ts_customer_092;
create regular tablespace ts_customer_092 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V2CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_093 of D1

drop tablespace ts_customer_093;
create regular tablespace ts_customer_093 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V3CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_094 of D1

drop tablespace ts_customer_094;
create regular tablespace ts_customer_094 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V4CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_095 of D1

drop tablespace ts_customer_095;
create regular tablespace ts_customer_095 pagesize 4K
  managed by database
  using

```

```

  (
    device '/dev/rD1F07V5CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_096 of D1

drop tablespace ts_customer_096;
create regular tablespace ts_customer_096 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V6CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_097 of D1

drop tablespace ts_customer_097;
create regular tablespace ts_customer_097 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V7CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_098 of D1

drop tablespace ts_customer_098;
create regular tablespace ts_customer_098 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V8CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_099 of D1

drop tablespace ts_customer_099;
create regular tablespace ts_customer_099 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V9CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_100 of D1

drop tablespace ts_customer_100;
create regular tablespace ts_customer_100 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V10CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_101 of D1

```

```

drop tablespace ts_customer_101;
create regular tablespace ts_customer_101 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V11CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_102 of D1

drop tablespace ts_customer_102;
create regular tablespace ts_customer_102 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V12CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_103 of D1

drop tablespace ts_customer_103;
create regular tablespace ts_customer_103 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V13CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_104 of D1

drop tablespace ts_customer_104;
create regular tablespace ts_customer_104 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V14CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_105 of D1

drop tablespace ts_customer_105;
create regular tablespace ts_customer_105 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V15CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_106 of D1

drop tablespace ts_customer_106;
create regular tablespace ts_customer_106 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V1CST' 5463808
  )

```

```

  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_107 of D1

drop tablespace ts_customer_107;
create regular tablespace ts_customer_107 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V2CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_108 of D1

drop tablespace ts_customer_108;
create regular tablespace ts_customer_108 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V3CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_109 of D1

drop tablespace ts_customer_109;
create regular tablespace ts_customer_109 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V4CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_110 of D1

drop tablespace ts_customer_110;
create regular tablespace ts_customer_110 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V5CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_111 of D1

drop tablespace ts_customer_111;
create regular tablespace ts_customer_111 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V6CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_112 of D1

drop tablespace ts_customer_112;
create regular tablespace ts_customer_112 pagesize 4K

```

```

  managed by database
  using
  (
    device '/dev/rD1F08V7CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_113 of D1

drop tablespace ts_customer_113;
create regular tablespace ts_customer_113 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V8CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_114 of D1

drop tablespace ts_customer_114;
create regular tablespace ts_customer_114 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V9CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_115 of D1

drop tablespace ts_customer_115;
create regular tablespace ts_customer_115 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V10CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_116 of D1

drop tablespace ts_customer_116;
create regular tablespace ts_customer_116 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V11CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_customer_117 of D1

drop tablespace ts_customer_117;
create regular tablespace ts_customer_117 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V12CST' 5463808
  )
  extentsize 256
  prefetchsize 4096;
commit;

```



```

-- now creating TS for ts_customer_118 of D1

drop tablespace ts_customer_118;
create regular tablespace ts_customer_118 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V13CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_119 of D1

drop tablespace ts_customer_119;
create regular tablespace ts_customer_119 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V14CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_customer_120 of D1

drop tablespace ts_customer_120;
create regular tablespace ts_customer_120 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V15CST' 5463808
)
extentsize 256
prefetchsize 4096;
commit;

connect reset;

```

### ts/crts\_dist.ddl

```

connect to tpcc;
-- now creating TS for ts_dist_001 of D1

drop tablespace ts_dist_001;
create regular tablespace ts_dist_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_002 of D1

drop tablespace ts_dist_002;
create regular tablespace ts_dist_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_dist_003 of D1

drop tablespace ts_dist_003;
create regular tablespace ts_dist_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_004 of D1

drop tablespace ts_dist_004;
create regular tablespace ts_dist_004 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V4DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_005 of D1

drop tablespace ts_dist_005;
create regular tablespace ts_dist_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_006 of D1

drop tablespace ts_dist_006;
create regular tablespace ts_dist_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V6DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_007 of D1

drop tablespace ts_dist_007;
create regular tablespace ts_dist_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_008 of D1

drop tablespace ts_dist_008;
create regular tablespace ts_dist_008 pagesize 4K
managed by database
using
(

```

```

    device '/dev/rD1F01V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_009 of D1

drop tablespace ts_dist_009;
create regular tablespace ts_dist_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_010 of D1

drop tablespace ts_dist_010;
create regular tablespace ts_dist_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_011 of D1

drop tablespace ts_dist_011;
create regular tablespace ts_dist_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_012 of D1

drop tablespace ts_dist_012;
create regular tablespace ts_dist_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_013 of D1

drop tablespace ts_dist_013;
create regular tablespace ts_dist_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_014 of D1

```

```

drop tablespace ts_dist_014;
create regular tablespace ts_dist_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_015 of D1

drop tablespace ts_dist_015;
create regular tablespace ts_dist_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V15DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_016 of D1

drop tablespace ts_dist_016;
create regular tablespace ts_dist_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_017 of D1

drop tablespace ts_dist_017;
create regular tablespace ts_dist_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_018 of D1

drop tablespace ts_dist_018;
create regular tablespace ts_dist_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_019 of D1

drop tablespace ts_dist_019;
create regular tablespace ts_dist_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4DIST' 640
)
extentsize 64

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_dist_020 of D1

drop tablespace ts_dist_020;
create regular tablespace ts_dist_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_021 of D1

drop tablespace ts_dist_021;
create regular tablespace ts_dist_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V6DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_022 of D1

drop tablespace ts_dist_022;
create regular tablespace ts_dist_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_023 of D1

drop tablespace ts_dist_023;
create regular tablespace ts_dist_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_024 of D1

drop tablespace ts_dist_024;
create regular tablespace ts_dist_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_025 of D1

drop tablespace ts_dist_025;
create regular tablespace ts_dist_025 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F02V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_026 of D1

drop tablespace ts_dist_026;
create regular tablespace ts_dist_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_027 of D1

drop tablespace ts_dist_027;
create regular tablespace ts_dist_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_028 of D1

drop tablespace ts_dist_028;
create regular tablespace ts_dist_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_029 of D1

drop tablespace ts_dist_029;
create regular tablespace ts_dist_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_030 of D1

drop tablespace ts_dist_030;
create regular tablespace ts_dist_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V15DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_dist_031 of D1

drop tablespace ts_dist_031;
create regular tablespace ts_dist_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_032 of D1

drop tablespace ts_dist_032;
create regular tablespace ts_dist_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_033 of D1

drop tablespace ts_dist_033;
create regular tablespace ts_dist_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_034 of D1

drop tablespace ts_dist_034;
create regular tablespace ts_dist_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_035 of D1

drop tablespace ts_dist_035;
create regular tablespace ts_dist_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_036 of D1

drop tablespace ts_dist_036;
create regular tablespace ts_dist_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V6DIST' 640

```

```

)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_037 of D1

drop tablespace ts_dist_037;
create regular tablespace ts_dist_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_038 of D1

drop tablespace ts_dist_038;
create regular tablespace ts_dist_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_039 of D1

drop tablespace ts_dist_039;
create regular tablespace ts_dist_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_040 of D1

drop tablespace ts_dist_040;
create regular tablespace ts_dist_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_041 of D1

drop tablespace ts_dist_041;
create regular tablespace ts_dist_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_042 of D1

drop tablespace ts_dist_042;

```

```

create regular tablespace ts_dist_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_043 of D1

drop tablespace ts_dist_043;
create regular tablespace ts_dist_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_044 of D1

drop tablespace ts_dist_044;
create regular tablespace ts_dist_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_045 of D1

drop tablespace ts_dist_045;
create regular tablespace ts_dist_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V15DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_046 of D1

drop tablespace ts_dist_046;
create regular tablespace ts_dist_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_047 of D1

drop tablespace ts_dist_047;
create regular tablespace ts_dist_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2DIST' 640
)
extentsize 64
prefetchsize 4096;

```

```

commit;
-- now creating TS for ts_dist_048 of D1
drop tablespace ts_dist_048;
create regular tablespace ts_dist_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_049 of D1
drop tablespace ts_dist_049;
create regular tablespace ts_dist_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_050 of D1
drop tablespace ts_dist_050;
create regular tablespace ts_dist_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_051 of D1
drop tablespace ts_dist_051;
create regular tablespace ts_dist_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V6DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_052 of D1
drop tablespace ts_dist_052;
create regular tablespace ts_dist_052 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_053 of D1
drop tablespace ts_dist_053;
create regular tablespace ts_dist_053 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F04V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_054 of D1
drop tablespace ts_dist_054;
create regular tablespace ts_dist_054 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_055 of D1
drop tablespace ts_dist_055;
create regular tablespace ts_dist_055 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_056 of D1
drop tablespace ts_dist_056;
create regular tablespace ts_dist_056 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_057 of D1
drop tablespace ts_dist_057;
create regular tablespace ts_dist_057 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_058 of D1
drop tablespace ts_dist_058;
create regular tablespace ts_dist_058 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_059 of D1

```

```

drop tablespace ts_dist_059;
create regular tablespace ts_dist_059 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_060 of D1
drop tablespace ts_dist_060;
create regular tablespace ts_dist_060 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V15DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_061 of D1
drop tablespace ts_dist_061;
create regular tablespace ts_dist_061 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_062 of D1
drop tablespace ts_dist_062;
create regular tablespace ts_dist_062 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_063 of D1
drop tablespace ts_dist_063;
create regular tablespace ts_dist_063 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_064 of D1
drop tablespace ts_dist_064;
create regular tablespace ts_dist_064 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4DIST' 640
)

```

```

        extentsize 64
        prefetchsize 4096;
commit;
-- now creating TS for ts_dist_065 of D1

drop tablespace ts_dist_065;
create regular tablespace ts_dist_065 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_066 of D1

drop tablespace ts_dist_066;
create regular tablespace ts_dist_066 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V6DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_067 of D1

drop tablespace ts_dist_067;
create regular tablespace ts_dist_067 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_068 of D1

drop tablespace ts_dist_068;
create regular tablespace ts_dist_068 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_069 of D1

drop tablespace ts_dist_069;
create regular tablespace ts_dist_069 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_070 of D1

drop tablespace ts_dist_070;
create regular tablespace ts_dist_070 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F05V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_071 of D1

drop tablespace ts_dist_071;
create regular tablespace ts_dist_071 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_072 of D1

drop tablespace ts_dist_072;
create regular tablespace ts_dist_072 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_073 of D1

drop tablespace ts_dist_073;
create regular tablespace ts_dist_073 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_074 of D1

drop tablespace ts_dist_074;
create regular tablespace ts_dist_074 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_075 of D1

drop tablespace ts_dist_075;
create regular tablespace ts_dist_075 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V15DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_dist_076 of D1

drop tablespace ts_dist_076;
create regular tablespace ts_dist_076 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_077 of D1

drop tablespace ts_dist_077;
create regular tablespace ts_dist_077 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_078 of D1

drop tablespace ts_dist_078;
create regular tablespace ts_dist_078 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_079 of D1

drop tablespace ts_dist_079;
create regular tablespace ts_dist_079 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_080 of D1

drop tablespace ts_dist_080;
create regular tablespace ts_dist_080 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;
-- now creating TS for ts_dist_081 of D1

drop tablespace ts_dist_081;
create regular tablespace ts_dist_081 pagesize 4K
managed by database
using
(

```

```

        device '/dev/rD1F06V6DIST' 640
    )
    extentsize 64
    prefetchsize 4096;
commit;

-- now creating TS for ts_dist_082 of D1

drop tablespace ts_dist_082;
create regular tablespace ts_dist_082 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_083 of D1

drop tablespace ts_dist_083;
create regular tablespace ts_dist_083 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_084 of D1

drop tablespace ts_dist_084;
create regular tablespace ts_dist_084 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_085 of D1

drop tablespace ts_dist_085;
create regular tablespace ts_dist_085 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_086 of D1

drop tablespace ts_dist_086;
create regular tablespace ts_dist_086 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_087 of D1

```

```

drop tablespace ts_dist_087;
create regular tablespace ts_dist_087 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_088 of D1

drop tablespace ts_dist_088;
create regular tablespace ts_dist_088 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_089 of D1

drop tablespace ts_dist_089;
create regular tablespace ts_dist_089 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_090 of D1

drop tablespace ts_dist_090;
create regular tablespace ts_dist_090 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V15DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_091 of D1

drop tablespace ts_dist_091;
create regular tablespace ts_dist_091 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_092 of D1

drop tablespace ts_dist_092;
create regular tablespace ts_dist_092 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2DIST' 640
)
extentsize 64

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_dist_093 of D1

drop tablespace ts_dist_093;
create regular tablespace ts_dist_093 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_094 of D1

drop tablespace ts_dist_094;
create regular tablespace ts_dist_094 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_095 of D1

drop tablespace ts_dist_095;
create regular tablespace ts_dist_095 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_096 of D1

drop tablespace ts_dist_096;
create regular tablespace ts_dist_096 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V6DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_097 of D1

drop tablespace ts_dist_097;
create regular tablespace ts_dist_097 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_098 of D1

drop tablespace ts_dist_098;
create regular tablespace ts_dist_098 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F07V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_099 of D1

drop tablespace ts_dist_099;
create regular tablespace ts_dist_099 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_100 of D1

drop tablespace ts_dist_100;
create regular tablespace ts_dist_100 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_101 of D1

drop tablespace ts_dist_101;
create regular tablespace ts_dist_101 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_102 of D1

drop tablespace ts_dist_102;
create regular tablespace ts_dist_102 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_103 of D1

drop tablespace ts_dist_103;
create regular tablespace ts_dist_103 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_dist_104 of D1

drop tablespace ts_dist_104;
create regular tablespace ts_dist_104 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_105 of D1

drop tablespace ts_dist_105;
create regular tablespace ts_dist_105 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V15DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_106 of D1

drop tablespace ts_dist_106;
create regular tablespace ts_dist_106 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_107 of D1

drop tablespace ts_dist_107;
create regular tablespace ts_dist_107 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_108 of D1

drop tablespace ts_dist_108;
create regular tablespace ts_dist_108 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_109 of D1

drop tablespace ts_dist_109;
create regular tablespace ts_dist_109 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4DIST' 640
)

```

```

)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_110 of D1

drop tablespace ts_dist_110;
create regular tablespace ts_dist_110 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_111 of D1

drop tablespace ts_dist_111;
create regular tablespace ts_dist_111 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V6DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_112 of D1

drop tablespace ts_dist_112;
create regular tablespace ts_dist_112 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V7DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_113 of D1

drop tablespace ts_dist_113;
create regular tablespace ts_dist_113 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V8DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_114 of D1

drop tablespace ts_dist_114;
create regular tablespace ts_dist_114 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V9DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_115 of D1

drop tablespace ts_dist_115;

```

```

create regular tablespace ts_dist_115 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V10DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_116 of D1

drop tablespace ts_dist_116;
create regular tablespace ts_dist_116 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V11DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_117 of D1

drop tablespace ts_dist_117;
create regular tablespace ts_dist_117 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V12DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_118 of D1

drop tablespace ts_dist_118;
create regular tablespace ts_dist_118 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V13DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_119 of D1

drop tablespace ts_dist_119;
create regular tablespace ts_dist_119 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V14DIST' 640
)
extentsize 64
prefetchsize 4096;
commit;

-- now creating TS for ts_dist_120 of D1

drop tablespace ts_dist_120;
create regular tablespace ts_dist_120 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V15DIST' 640
)
extentsize 64
prefetchsize 4096;

```

```

commit;
connect reset;

ts/crts_history.ddl

connect to tpcc;
-- now creating TS for ts_history_001 of D1

drop tablespace ts_history_001;
create regular tablespace ts_history_001 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V1HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_002 of D1

drop tablespace ts_history_002;
create regular tablespace ts_history_002 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_003 of D1

drop tablespace ts_history_003;
create regular tablespace ts_history_003 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_004 of D1

drop tablespace ts_history_004;
create regular tablespace ts_history_004 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V4HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_005 of D1

drop tablespace ts_history_005;
create regular tablespace ts_history_005 pagesize 16K
managed by database
using
(

```

```

    device '/dev/rD1F01V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_006 of D1

drop tablespace ts_history_006;
create regular tablespace ts_history_006 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V6HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_007 of D1

drop tablespace ts_history_007;
create regular tablespace ts_history_007 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_008 of D1

drop tablespace ts_history_008;
create regular tablespace ts_history_008 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V8HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_009 of D1

drop tablespace ts_history_009;
create regular tablespace ts_history_009 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_010 of D1

drop tablespace ts_history_010;
create regular tablespace ts_history_010 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V10HIST' 135168
)
extentsize 256

```



```

        prefetchsize 4096
        bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_011 of D1

drop tablespace ts_history_011;
create regular tablespace ts_history_011 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_012 of D1

drop tablespace ts_history_012;
create regular tablespace ts_history_012 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V12HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_013 of D1

drop tablespace ts_history_013;
create regular tablespace ts_history_013 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_014 of D1

drop tablespace ts_history_014;
create regular tablespace ts_history_014 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V14HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_015 of D1

drop tablespace ts_history_015;
create regular tablespace ts_history_015 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V15HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_016 of D1

drop tablespace ts_history_016;
create regular tablespace ts_history_016 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V1HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_017 of D1

drop tablespace ts_history_017;
create regular tablespace ts_history_017 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_018 of D1

drop tablespace ts_history_018;
create regular tablespace ts_history_018 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_019 of D1

drop tablespace ts_history_019;
create regular tablespace ts_history_019 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V4HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_020 of D1

drop tablespace ts_history_020;
create regular tablespace ts_history_020 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_021 of D1

```

```

drop tablespace ts_history_021;
create regular tablespace ts_history_021 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V6HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_022 of D1

drop tablespace ts_history_022;
create regular tablespace ts_history_022 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_023 of D1

drop tablespace ts_history_023;
create regular tablespace ts_history_023 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V8HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_024 of D1

drop tablespace ts_history_024;
create regular tablespace ts_history_024 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_025 of D1

drop tablespace ts_history_025;
create regular tablespace ts_history_025 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V10HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_026 of D1

drop tablespace ts_history_026;
create regular tablespace ts_history_026 pagesize 16K
managed by database

```

```

using
(
    device '/dev/rD1F02V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_027 of D1

drop tablespace ts_history_027;
create regular tablespace ts_history_027 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V12HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_028 of D1

drop tablespace ts_history_028;
create regular tablespace ts_history_028 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_029 of D1

drop tablespace ts_history_029;
create regular tablespace ts_history_029 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V14HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_030 of D1

drop tablespace ts_history_030;
create regular tablespace ts_history_030 pagesize 16K
managed by database
using
(
    device '/dev/rD1F02V15HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_031 of D1

drop tablespace ts_history_031;
create regular tablespace ts_history_031 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V1HIST' 135168

```

```

)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_032 of D1

drop tablespace ts_history_032;
create regular tablespace ts_history_032 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_033 of D1

drop tablespace ts_history_033;
create regular tablespace ts_history_033 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_034 of D1

drop tablespace ts_history_034;
create regular tablespace ts_history_034 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V4HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_035 of D1

drop tablespace ts_history_035;
create regular tablespace ts_history_035 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_036 of D1

drop tablespace ts_history_036;
create regular tablespace ts_history_036 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V6HIST' 135168
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_037 of D1

drop tablespace ts_history_037;
create regular tablespace ts_history_037 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_038 of D1

drop tablespace ts_history_038;
create regular tablespace ts_history_038 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V8HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_039 of D1

drop tablespace ts_history_039;
create regular tablespace ts_history_039 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_040 of D1

drop tablespace ts_history_040;
create regular tablespace ts_history_040 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V10HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_041 of D1

drop tablespace ts_history_041;
create regular tablespace ts_history_041 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_042 of D1
drop tablespace ts_history_042;
create regular tablespace ts_history_042 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V12HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_043 of D1
drop tablespace ts_history_043;
create regular tablespace ts_history_043 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_044 of D1
drop tablespace ts_history_044;
create regular tablespace ts_history_044 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V14HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_045 of D1
drop tablespace ts_history_045;
create regular tablespace ts_history_045 pagesize 16K
managed by database
using
(
    device '/dev/rD1F03V15HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_046 of D1
drop tablespace ts_history_046;
create regular tablespace ts_history_046 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V1HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_047 of D1
drop tablespace ts_history_047;

```

```

create regular tablespace ts_history_047 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_048 of D1
drop tablespace ts_history_048;
create regular tablespace ts_history_048 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_049 of D1
drop tablespace ts_history_049;
create regular tablespace ts_history_049 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V4HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_050 of D1
drop tablespace ts_history_050;
create regular tablespace ts_history_050 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_051 of D1
drop tablespace ts_history_051;
create regular tablespace ts_history_051 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V6HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_052 of D1
drop tablespace ts_history_052;
create regular tablespace ts_history_052 pagesize 16K
managed by database
using

```

```

(
    device '/dev/rD1F04V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_053 of D1
drop tablespace ts_history_053;
create regular tablespace ts_history_053 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V8HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_054 of D1
drop tablespace ts_history_054;
create regular tablespace ts_history_054 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_055 of D1
drop tablespace ts_history_055;
create regular tablespace ts_history_055 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V10HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_056 of D1
drop tablespace ts_history_056;
create regular tablespace ts_history_056 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_057 of D1
drop tablespace ts_history_057;
create regular tablespace ts_history_057 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V12HIST' 135168
)

```

```

        extentsize 256
        prefetchsize 4096
        bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_058 of D1

drop tablespace ts_history_058;
create regular tablespace ts_history_058 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_059 of D1

drop tablespace ts_history_059;
create regular tablespace ts_history_059 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V14HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_060 of D1

drop tablespace ts_history_060;
create regular tablespace ts_history_060 pagesize 16K
managed by database
using
(
    device '/dev/rD1F04V15HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_061 of D1

drop tablespace ts_history_061;
create regular tablespace ts_history_061 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V1HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_062 of D1

drop tablespace ts_history_062;
create regular tablespace ts_history_062 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;

```

```

commit;

-- now creating TS for ts_history_063 of D1

drop tablespace ts_history_063;
create regular tablespace ts_history_063 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_064 of D1

drop tablespace ts_history_064;
create regular tablespace ts_history_064 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V4HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_065 of D1

drop tablespace ts_history_065;
create regular tablespace ts_history_065 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_066 of D1

drop tablespace ts_history_066;
create regular tablespace ts_history_066 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V6HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_067 of D1

drop tablespace ts_history_067;
create regular tablespace ts_history_067 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_068 of D1

```

```

drop tablespace ts_history_068;
create regular tablespace ts_history_068 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V8HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_069 of D1

drop tablespace ts_history_069;
create regular tablespace ts_history_069 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_070 of D1

drop tablespace ts_history_070;
create regular tablespace ts_history_070 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V10HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_071 of D1

drop tablespace ts_history_071;
create regular tablespace ts_history_071 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_072 of D1

drop tablespace ts_history_072;
create regular tablespace ts_history_072 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V12HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_073 of D1

drop tablespace ts_history_073;
create regular tablespace ts_history_073 pagesize 16K

```

```

managed by database
using
(
    device '/dev/rD1F05V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_074 of D1

drop tablespace ts_history_074;
create regular tablespace ts_history_074 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V14HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_075 of D1

drop tablespace ts_history_075;
create regular tablespace ts_history_075 pagesize 16K
managed by database
using
(
    device '/dev/rD1F05V15HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_076 of D1

drop tablespace ts_history_076;
create regular tablespace ts_history_076 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V1HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_077 of D1

drop tablespace ts_history_077;
create regular tablespace ts_history_077 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_078 of D1

drop tablespace ts_history_078;
create regular tablespace ts_history_078 pagesize 16K
managed by database
using
(

```

```

    device '/dev/rD1F06V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_079 of D1

drop tablespace ts_history_079;
create regular tablespace ts_history_079 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V4HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_080 of D1

drop tablespace ts_history_080;
create regular tablespace ts_history_080 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_081 of D1

drop tablespace ts_history_081;
create regular tablespace ts_history_081 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V6HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_082 of D1

drop tablespace ts_history_082;
create regular tablespace ts_history_082 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_083 of D1

drop tablespace ts_history_083;
create regular tablespace ts_history_083 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V8HIST' 135168
)
extentsize 256

```

```

    prefetchsize 4096
    bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_084 of D1

drop tablespace ts_history_084;
create regular tablespace ts_history_084 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_085 of D1

drop tablespace ts_history_085;
create regular tablespace ts_history_085 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V10HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_086 of D1

drop tablespace ts_history_086;
create regular tablespace ts_history_086 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_087 of D1

drop tablespace ts_history_087;
create regular tablespace ts_history_087 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V12HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_088 of D1

drop tablespace ts_history_088;
create regular tablespace ts_history_088 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_089 of D1

drop tablespace ts_history_089;
create regular tablespace ts_history_089 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V14HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_090 of D1

drop tablespace ts_history_090;
create regular tablespace ts_history_090 pagesize 16K
managed by database
using
(
    device '/dev/rD1F06V15HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_091 of D1

drop tablespace ts_history_091;
create regular tablespace ts_history_091 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V1HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_092 of D1

drop tablespace ts_history_092;
create regular tablespace ts_history_092 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_093 of D1

drop tablespace ts_history_093;
create regular tablespace ts_history_093 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_094 of D1

```

```

drop tablespace ts_history_094;
create regular tablespace ts_history_094 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V4HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_095 of D1

drop tablespace ts_history_095;
create regular tablespace ts_history_095 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_096 of D1

drop tablespace ts_history_096;
create regular tablespace ts_history_096 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V6HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_097 of D1

drop tablespace ts_history_097;
create regular tablespace ts_history_097 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_098 of D1

drop tablespace ts_history_098;
create regular tablespace ts_history_098 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V8HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_099 of D1

drop tablespace ts_history_099;
create regular tablespace ts_history_099 pagesize 16K
managed by database

```

```

using
(
    device '/dev/rD1F07V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_100 of D1

drop tablespace ts_history_100;
create regular tablespace ts_history_100 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V10HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_101 of D1

drop tablespace ts_history_101;
create regular tablespace ts_history_101 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_102 of D1

drop tablespace ts_history_102;
create regular tablespace ts_history_102 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V12HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_103 of D1

drop tablespace ts_history_103;
create regular tablespace ts_history_103 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_104 of D1

drop tablespace ts_history_104;
create regular tablespace ts_history_104 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V14HIST' 135168
)

```

```

    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_105 of D1

drop tablespace ts_history_105;
create regular tablespace ts_history_105 pagesize 16K
managed by database
using
(
    device '/dev/rD1F07V15HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_106 of D1

drop tablespace ts_history_106;
create regular tablespace ts_history_106 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V1HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_107 of D1

drop tablespace ts_history_107;
create regular tablespace ts_history_107 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V2HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_108 of D1

drop tablespace ts_history_108;
create regular tablespace ts_history_108 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V3HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_109 of D1

drop tablespace ts_history_109;
create regular tablespace ts_history_109 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V4HIST' 135168
)
extentsize 256
prefetchsize 4096

```

```

    bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_110 of D1

drop tablespace ts_history_110;
create regular tablespace ts_history_110 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V5HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_111 of D1

drop tablespace ts_history_111;
create regular tablespace ts_history_111 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V6HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_112 of D1

drop tablespace ts_history_112;
create regular tablespace ts_history_112 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V7HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_113 of D1

drop tablespace ts_history_113;
create regular tablespace ts_history_113 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V8HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_114 of D1

drop tablespace ts_history_114;
create regular tablespace ts_history_114 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V9HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_115 of D1

drop tablespace ts_history_115;
create regular tablespace ts_history_115 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V10HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_116 of D1

drop tablespace ts_history_116;
create regular tablespace ts_history_116 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V11HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_117 of D1

drop tablespace ts_history_117;
create regular tablespace ts_history_117 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V12HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_118 of D1

drop tablespace ts_history_118;
create regular tablespace ts_history_118 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V13HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_119 of D1

drop tablespace ts_history_119;
create regular tablespace ts_history_119 pagesize 16K
managed by database
using
(
    device '/dev/rD1F08V14HIST' 135168
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_120 of D1

drop tablespace ts_history_120;

```

```

create regular tablespace ts_history_120 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F08V15HIST' 135168
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp16K;
commit;

connect reset;

```

### ts/crts\_item.ddl

```

connect to tpcc;
-- now creating TS for ts_item_001 of D1

```

```

drop tablespace ts_item_001;
create regular tablespace ts_item_001 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F01V1ITEM' 1408,
    device '/dev/rD1F01V2ITEM' 1408,
    device '/dev/rD1F01V3ITEM' 1408,
    device '/dev/rD1F01V4ITEM' 1408,
    device '/dev/rD1F01V5ITEM' 1408,
    device '/dev/rD1F01V6ITEM' 1408,
    device '/dev/rD1F01V7ITEM' 1408,
    device '/dev/rD1F01V8ITEM' 1408,
    device '/dev/rD1F01V9ITEM' 1408,
    device '/dev/rD1F01V10ITEM' 1408,
    device '/dev/rD1F01V11ITEM' 1408,
    device '/dev/rD1F01V12ITEM' 1408,
    device '/dev/rD1F01V13ITEM' 1408,
    device '/dev/rD1F01V14ITEM' 1408,
    device '/dev/rD1F01V15ITEM' 1408,
    device '/dev/rD1F02V1ITEM' 1408,
    device '/dev/rD1F02V2ITEM' 1408,
    device '/dev/rD1F02V3ITEM' 1408,
    device '/dev/rD1F02V4ITEM' 1408,
    device '/dev/rD1F02V5ITEM' 1408,
    device '/dev/rD1F02V6ITEM' 1408,
    device '/dev/rD1F02V7ITEM' 1408,
    device '/dev/rD1F02V8ITEM' 1408,
    device '/dev/rD1F02V9ITEM' 1408,
    device '/dev/rD1F02V10ITEM' 1408,
    device '/dev/rD1F02V11ITEM' 1408,
    device '/dev/rD1F02V12ITEM' 1408,
    device '/dev/rD1F02V13ITEM' 1408,
    device '/dev/rD1F02V14ITEM' 1408,
    device '/dev/rD1F02V15ITEM' 1408,
    device '/dev/rD1F03V1ITEM' 1408,
    device '/dev/rD1F03V2ITEM' 1408,
    device '/dev/rD1F03V3ITEM' 1408,
    device '/dev/rD1F03V4ITEM' 1408,
    device '/dev/rD1F03V5ITEM' 1408,
    device '/dev/rD1F03V6ITEM' 1408,
    device '/dev/rD1F03V7ITEM' 1408,
    device '/dev/rD1F03V8ITEM' 1408,
    device '/dev/rD1F03V9ITEM' 1408,
    device '/dev/rD1F03V10ITEM' 1408,
    device '/dev/rD1F03V11ITEM' 1408,
    device '/dev/rD1F03V12ITEM' 1408,
    device '/dev/rD1F03V13ITEM' 1408,
    device '/dev/rD1F03V14ITEM' 1408,
    device '/dev/rD1F03V15ITEM' 1408,
    device '/dev/rD1F04V1ITEM' 1408,
    device '/dev/rD1F04V2ITEM' 1408,

```

```

device '/dev/rD1F04V3ITEM' 1408,
device '/dev/rD1F04V4ITEM' 1408,
device '/dev/rD1F04V5ITEM' 1408,
device '/dev/rD1F04V6ITEM' 1408,
device '/dev/rD1F04V7ITEM' 1408,
device '/dev/rD1F04V8ITEM' 1408,
device '/dev/rD1F04V9ITEM' 1408,
device '/dev/rD1F04V10ITEM' 1408,
device '/dev/rD1F04V11ITEM' 1408,
device '/dev/rD1F04V12ITEM' 1408,
device '/dev/rD1F04V13ITEM' 1408,
device '/dev/rD1F04V14ITEM' 1408,
device '/dev/rD1F04V15ITEM' 1408,
device '/dev/rD1F05V1ITEM' 1408,
device '/dev/rD1F05V2ITEM' 1408,
device '/dev/rD1F05V3ITEM' 1408,
device '/dev/rD1F05V4ITEM' 1408,
device '/dev/rD1F05V5ITEM' 1408,
device '/dev/rD1F05V6ITEM' 1408,
device '/dev/rD1F05V7ITEM' 1408,
device '/dev/rD1F05V8ITEM' 1408,
device '/dev/rD1F05V9ITEM' 1408,
device '/dev/rD1F05V10ITEM' 1408,
device '/dev/rD1F05V11ITEM' 1408,
device '/dev/rD1F05V12ITEM' 1408,
device '/dev/rD1F05V13ITEM' 1408,
device '/dev/rD1F05V14ITEM' 1408,
device '/dev/rD1F05V15ITEM' 1408,
device '/dev/rD1F06V1ITEM' 1408,
device '/dev/rD1F06V2ITEM' 1408,
device '/dev/rD1F06V3ITEM' 1408,
device '/dev/rD1F06V4ITEM' 1408,
device '/dev/rD1F06V5ITEM' 1408,
device '/dev/rD1F06V6ITEM' 1408,
device '/dev/rD1F06V7ITEM' 1408,
device '/dev/rD1F06V8ITEM' 1408,
device '/dev/rD1F06V9ITEM' 1408,
device '/dev/rD1F06V10ITEM' 1408,
device '/dev/rD1F06V11ITEM' 1408,
device '/dev/rD1F06V12ITEM' 1408,
device '/dev/rD1F06V13ITEM' 1408,
device '/dev/rD1F06V14ITEM' 1408,
device '/dev/rD1F06V15ITEM' 1408,
device '/dev/rD1F07V1ITEM' 1408,
device '/dev/rD1F07V2ITEM' 1408,
device '/dev/rD1F07V3ITEM' 1408,
device '/dev/rD1F07V4ITEM' 1408,
device '/dev/rD1F07V5ITEM' 1408,
device '/dev/rD1F07V6ITEM' 1408,
device '/dev/rD1F07V7ITEM' 1408,
device '/dev/rD1F07V8ITEM' 1408,
device '/dev/rD1F07V9ITEM' 1408,
device '/dev/rD1F07V10ITEM' 1408,
device '/dev/rD1F07V11ITEM' 1408,
device '/dev/rD1F07V12ITEM' 1408,
device '/dev/rD1F07V13ITEM' 1408,
device '/dev/rD1F07V14ITEM' 1408,
device '/dev/rD1F07V15ITEM' 1408,
device '/dev/rD1F08V1ITEM' 1408,
device '/dev/rD1F08V2ITEM' 1408,
device '/dev/rD1F08V3ITEM' 1408,
device '/dev/rD1F08V4ITEM' 1408,
device '/dev/rD1F08V5ITEM' 1408,
device '/dev/rD1F08V6ITEM' 1408,
device '/dev/rD1F08V7ITEM' 1408,
device '/dev/rD1F08V8ITEM' 1408,
device '/dev/rD1F08V9ITEM' 1408,
device '/dev/rD1F08V10ITEM' 1408,
device '/dev/rD1F08V11ITEM' 1408,
device '/dev/rD1F08V12ITEM' 1408,
device '/dev/rD1F08V13ITEM' 1408,
device '/dev/rD1F08V14ITEM' 1408,
device '/dev/rD1F08V15ITEM' 1408

```

```

)
  extentsize 16
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

### ts/crts\_neworda.ddl

```

connect to tpcc;
-- now creating TS for ts_neworda_001 of D1

```

```

drop tablespace ts_neworda_001;
create regular tablespace ts_neworda_001 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_002 of D1

```

```

drop tablespace ts_neworda_002;
create regular tablespace ts_neworda_002 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V2NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_003 of D1

```

```

drop tablespace ts_neworda_003;
create regular tablespace ts_neworda_003 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V3NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_004 of D1

```

```

drop tablespace ts_neworda_004;
create regular tablespace ts_neworda_004 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V4NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_005 of D1

```

```

drop tablespace ts_neworda_005;
create regular tablespace ts_neworda_005 pagesize 4K
  managed by database
  using
  (

```



```

        device '/dev/rD1F01V5NORA' 88320
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_006 of D1

drop tablespace ts_neworda_006;
create regular tablespace ts_neworda_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V6NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_007 of D1

drop tablespace ts_neworda_007;
create regular tablespace ts_neworda_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V7NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_008 of D1

drop tablespace ts_neworda_008;
create regular tablespace ts_neworda_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V8NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_009 of D1

drop tablespace ts_neworda_009;
create regular tablespace ts_neworda_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V9NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_010 of D1

drop tablespace ts_neworda_010;
create regular tablespace ts_neworda_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V10NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_011 of D1

```

```

drop tablespace ts_neworda_011;
create regular tablespace ts_neworda_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V11NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_012 of D1

drop tablespace ts_neworda_012;
create regular tablespace ts_neworda_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V12NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_013 of D1

drop tablespace ts_neworda_013;
create regular tablespace ts_neworda_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V13NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_014 of D1

drop tablespace ts_neworda_014;
create regular tablespace ts_neworda_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V14NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_015 of D1

drop tablespace ts_neworda_015;
create regular tablespace ts_neworda_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V15NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_016 of D1

drop tablespace ts_neworda_016;
create regular tablespace ts_neworda_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1NORA' 88320
)
extentsize 256

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_017 of D1

drop tablespace ts_neworda_017;
create regular tablespace ts_neworda_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_018 of D1

drop tablespace ts_neworda_018;
create regular tablespace ts_neworda_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_019 of D1

drop tablespace ts_neworda_019;
create regular tablespace ts_neworda_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_020 of D1

drop tablespace ts_neworda_020;
create regular tablespace ts_neworda_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_021 of D1

drop tablespace ts_neworda_021;
create regular tablespace ts_neworda_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V6NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_022 of D1

drop tablespace ts_neworda_022;
create regular tablespace ts_neworda_022 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F02V7NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_023 of D1

drop tablespace ts_neworda_023;
create regular tablespace ts_neworda_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V8NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_024 of D1

drop tablespace ts_neworda_024;
create regular tablespace ts_neworda_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V9NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_025 of D1

drop tablespace ts_neworda_025;
create regular tablespace ts_neworda_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V10NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_026 of D1

drop tablespace ts_neworda_026;
create regular tablespace ts_neworda_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V11NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_027 of D1

drop tablespace ts_neworda_027;
create regular tablespace ts_neworda_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V12NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_028 of D1

drop tablespace ts_neworda_028;
create regular tablespace ts_neworda_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V13NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_029 of D1

drop tablespace ts_neworda_029;
create regular tablespace ts_neworda_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V14NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_030 of D1

drop tablespace ts_neworda_030;
create regular tablespace ts_neworda_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V15NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_031 of D1

drop tablespace ts_neworda_031;
create regular tablespace ts_neworda_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_032 of D1

drop tablespace ts_neworda_032;
create regular tablespace ts_neworda_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_033 of D1

drop tablespace ts_neworda_033;
create regular tablespace ts_neworda_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3NORA' 88320
)

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_034 of D1

drop tablespace ts_neworda_034;
create regular tablespace ts_neworda_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_035 of D1

drop tablespace ts_neworda_035;
create regular tablespace ts_neworda_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_036 of D1

drop tablespace ts_neworda_036;
create regular tablespace ts_neworda_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V6NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_037 of D1

drop tablespace ts_neworda_037;
create regular tablespace ts_neworda_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V7NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_038 of D1

drop tablespace ts_neworda_038;
create regular tablespace ts_neworda_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V8NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_039 of D1

drop tablespace ts_neworda_039;

```

```

create regular tablespace ts_neworda_039 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V9NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_040 of D1

drop tablespace ts_neworda_040;
create regular tablespace ts_neworda_040 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V10NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_041 of D1

drop tablespace ts_neworda_041;
create regular tablespace ts_neworda_041 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V11NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_042 of D1

drop tablespace ts_neworda_042;
create regular tablespace ts_neworda_042 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V12NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_043 of D1

drop tablespace ts_neworda_043;
create regular tablespace ts_neworda_043 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V13NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_044 of D1

drop tablespace ts_neworda_044;
create regular tablespace ts_neworda_044 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V14NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_neworda_045 of D1

drop tablespace ts_neworda_045;
create regular tablespace ts_neworda_045 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V15NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_046 of D1

drop tablespace ts_neworda_046;
create regular tablespace ts_neworda_046 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_047 of D1

drop tablespace ts_neworda_047;
create regular tablespace ts_neworda_047 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V2NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_048 of D1

drop tablespace ts_neworda_048;
create regular tablespace ts_neworda_048 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V3NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_049 of D1

drop tablespace ts_neworda_049;
create regular tablespace ts_neworda_049 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V4NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_050 of D1

drop tablespace ts_neworda_050;
create regular tablespace ts_neworda_050 pagesize 4K
  managed by database
  using

```

```

  (
    device '/dev/rD1F04V5NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_051 of D1

drop tablespace ts_neworda_051;
create regular tablespace ts_neworda_051 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V6NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_052 of D1

drop tablespace ts_neworda_052;
create regular tablespace ts_neworda_052 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V7NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_053 of D1

drop tablespace ts_neworda_053;
create regular tablespace ts_neworda_053 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V8NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_054 of D1

drop tablespace ts_neworda_054;
create regular tablespace ts_neworda_054 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V9NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_055 of D1

drop tablespace ts_neworda_055;
create regular tablespace ts_neworda_055 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V10NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_056 of D1

```

```

drop tablespace ts_neworda_056;
create regular tablespace ts_neworda_056 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V11NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_057 of D1

drop tablespace ts_neworda_057;
create regular tablespace ts_neworda_057 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V12NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_058 of D1

drop tablespace ts_neworda_058;
create regular tablespace ts_neworda_058 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V13NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_059 of D1

drop tablespace ts_neworda_059;
create regular tablespace ts_neworda_059 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V14NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_060 of D1

drop tablespace ts_neworda_060;
create regular tablespace ts_neworda_060 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V15NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_061 of D1

drop tablespace ts_neworda_061;
create regular tablespace ts_neworda_061 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V1NORA' 88320
  )

```

```

  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_062 of D1

drop tablespace ts_neworda_062;
create regular tablespace ts_neworda_062 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V2NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_063 of D1

drop tablespace ts_neworda_063;
create regular tablespace ts_neworda_063 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V3NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_064 of D1

drop tablespace ts_neworda_064;
create regular tablespace ts_neworda_064 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V4NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_065 of D1

drop tablespace ts_neworda_065;
create regular tablespace ts_neworda_065 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V5NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_066 of D1

drop tablespace ts_neworda_066;
create regular tablespace ts_neworda_066 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V6NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_067 of D1

drop tablespace ts_neworda_067;
create regular tablespace ts_neworda_067 pagesize 4K

```

```

  managed by database
  using
  (
    device '/dev/rD1F05V7NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_068 of D1

drop tablespace ts_neworda_068;
create regular tablespace ts_neworda_068 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V8NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_069 of D1

drop tablespace ts_neworda_069;
create regular tablespace ts_neworda_069 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V9NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_070 of D1

drop tablespace ts_neworda_070;
create regular tablespace ts_neworda_070 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V10NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_071 of D1

drop tablespace ts_neworda_071;
create regular tablespace ts_neworda_071 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V11NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_072 of D1

drop tablespace ts_neworda_072;
create regular tablespace ts_neworda_072 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V12NORA' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_073 of D1

drop tablespace ts_neworda_073;
create regular tablespace ts_neworda_073 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V13NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_074 of D1

drop tablespace ts_neworda_074;
create regular tablespace ts_neworda_074 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V14NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_075 of D1

drop tablespace ts_neworda_075;
create regular tablespace ts_neworda_075 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V15NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_076 of D1

drop tablespace ts_neworda_076;
create regular tablespace ts_neworda_076 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_077 of D1

drop tablespace ts_neworda_077;
create regular tablespace ts_neworda_077 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_078 of D1

drop tablespace ts_neworda_078;
create regular tablespace ts_neworda_078 pagesize 4K
managed by database
using
(

```

```

    device '/dev/rD1F06V3NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_079 of D1

drop tablespace ts_neworda_079;
create regular tablespace ts_neworda_079 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_080 of D1

drop tablespace ts_neworda_080;
create regular tablespace ts_neworda_080 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_081 of D1

drop tablespace ts_neworda_081;
create regular tablespace ts_neworda_081 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V6NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_082 of D1

drop tablespace ts_neworda_082;
create regular tablespace ts_neworda_082 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V7NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_083 of D1

drop tablespace ts_neworda_083;
create regular tablespace ts_neworda_083 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V8NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_084 of D1

```

```

drop tablespace ts_neworda_084;
create regular tablespace ts_neworda_084 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V9NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_085 of D1

drop tablespace ts_neworda_085;
create regular tablespace ts_neworda_085 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V10NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_086 of D1

drop tablespace ts_neworda_086;
create regular tablespace ts_neworda_086 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V11NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_087 of D1

drop tablespace ts_neworda_087;
create regular tablespace ts_neworda_087 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V12NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_088 of D1

drop tablespace ts_neworda_088;
create regular tablespace ts_neworda_088 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V13NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworda_089 of D1

drop tablespace ts_neworda_089;
create regular tablespace ts_neworda_089 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V14NORA' 88320
)
extentsize 256

```

```

        prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_090 of D1
drop tablespace ts_neworda_090;
create regular tablespace ts_neworda_090 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V15NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_091 of D1
drop tablespace ts_neworda_091;
create regular tablespace ts_neworda_091 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_092 of D1
drop tablespace ts_neworda_092;
create regular tablespace ts_neworda_092 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_093 of D1
drop tablespace ts_neworda_093;
create regular tablespace ts_neworda_093 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_094 of D1
drop tablespace ts_neworda_094;
create regular tablespace ts_neworda_094 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_095 of D1
drop tablespace ts_neworda_095;
create regular tablespace ts_neworda_095 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F07V5NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_096 of D1
drop tablespace ts_neworda_096;
create regular tablespace ts_neworda_096 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V6NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_097 of D1
drop tablespace ts_neworda_097;
create regular tablespace ts_neworda_097 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V7NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_098 of D1
drop tablespace ts_neworda_098;
create regular tablespace ts_neworda_098 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V8NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_099 of D1
drop tablespace ts_neworda_099;
create regular tablespace ts_neworda_099 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V9NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_100 of D1
drop tablespace ts_neworda_100;
create regular tablespace ts_neworda_100 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V10NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_neworda_101 of D1
drop tablespace ts_neworda_101;
create regular tablespace ts_neworda_101 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V11NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_102 of D1
drop tablespace ts_neworda_102;
create regular tablespace ts_neworda_102 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V12NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_103 of D1
drop tablespace ts_neworda_103;
create regular tablespace ts_neworda_103 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V13NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_104 of D1
drop tablespace ts_neworda_104;
create regular tablespace ts_neworda_104 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V14NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_105 of D1
drop tablespace ts_neworda_105;
create regular tablespace ts_neworda_105 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V15NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_106 of D1
drop tablespace ts_neworda_106;
create regular tablespace ts_neworda_106 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1NORA' 88320
)

```

```

    )
    extentsize 256
    prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_107 of D1
drop tablespace ts_neworda_107;
create regular tablespace ts_neworda_107 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_108 of D1
drop tablespace ts_neworda_108;
create regular tablespace ts_neworda_108 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_109 of D1
drop tablespace ts_neworda_109;
create regular tablespace ts_neworda_109 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_110 of D1
drop tablespace ts_neworda_110;
create regular tablespace ts_neworda_110 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_111 of D1
drop tablespace ts_neworda_111;
create regular tablespace ts_neworda_111 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V6NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_112 of D1
drop tablespace ts_neworda_112;

```

```

create regular tablespace ts_neworda_112 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V7NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_113 of D1
drop tablespace ts_neworda_113;
create regular tablespace ts_neworda_113 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V8NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_114 of D1
drop tablespace ts_neworda_114;
create regular tablespace ts_neworda_114 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V9NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_115 of D1
drop tablespace ts_neworda_115;
create regular tablespace ts_neworda_115 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V10NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_116 of D1
drop tablespace ts_neworda_116;
create regular tablespace ts_neworda_116 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V11NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_117 of D1
drop tablespace ts_neworda_117;
create regular tablespace ts_neworda_117 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V12NORA' 88320
)
extentsize 256
prefetchsize 4096;

```

```

commit;
-- now creating TS for ts_neworda_118 of D1
drop tablespace ts_neworda_118;
create regular tablespace ts_neworda_118 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V13NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_119 of D1
drop tablespace ts_neworda_119;
create regular tablespace ts_neworda_119 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V14NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_neworda_120 of D1
drop tablespace ts_neworda_120;
create regular tablespace ts_neworda_120 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V15NORA' 88320
)
extentsize 256
prefetchsize 4096;
commit;
connect reset;

ts/crts_newordb.ddl

connect to tpc;
-- now creating TS for ts_newordb_001 of D1
drop tablespace ts_newordb_001;
create regular tablespace ts_newordb_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_002 of D1
drop tablespace ts_newordb_002;
create regular tablespace ts_newordb_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2NORB' 88320
)
extentsize 256
prefetchsize 4096;

```

```

commit;
-- now creating TS for ts_newordb_003 of D1
drop tablespace ts_newordb_003;
create regular tablespace ts_newordb_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_004 of D1
drop tablespace ts_newordb_004;
create regular tablespace ts_newordb_004 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V4NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_005 of D1
drop tablespace ts_newordb_005;
create regular tablespace ts_newordb_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_006 of D1
drop tablespace ts_newordb_006;
create regular tablespace ts_newordb_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V6NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_007 of D1
drop tablespace ts_newordb_007;
create regular tablespace ts_newordb_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V7NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_008 of D1
drop tablespace ts_newordb_008;
create regular tablespace ts_newordb_008 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F01V8NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_009 of D1
drop tablespace ts_newordb_009;
create regular tablespace ts_newordb_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V9NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_010 of D1
drop tablespace ts_newordb_010;
create regular tablespace ts_newordb_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V10NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_011 of D1
drop tablespace ts_newordb_011;
create regular tablespace ts_newordb_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V11NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_012 of D1
drop tablespace ts_newordb_012;
create regular tablespace ts_newordb_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V12NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_013 of D1
drop tablespace ts_newordb_013;
create regular tablespace ts_newordb_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V13NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_014 of D1

```

```

drop tablespace ts_newordb_014;
create regular tablespace ts_newordb_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V14NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_015 of D1
drop tablespace ts_newordb_015;
create regular tablespace ts_newordb_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V15NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_016 of D1
drop tablespace ts_newordb_016;
create regular tablespace ts_newordb_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_017 of D1
drop tablespace ts_newordb_017;
create regular tablespace ts_newordb_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_018 of D1
drop tablespace ts_newordb_018;
create regular tablespace ts_newordb_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_019 of D1
drop tablespace ts_newordb_019;
create regular tablespace ts_newordb_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4NORB' 88320
)

```



```

        extentsize 256
        prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_020 of D1
drop tablespace ts_newordb_020;
create regular tablespace ts_newordb_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_021 of D1
drop tablespace ts_newordb_021;
create regular tablespace ts_newordb_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V6NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_022 of D1
drop tablespace ts_newordb_022;
create regular tablespace ts_newordb_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V7NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_023 of D1
drop tablespace ts_newordb_023;
create regular tablespace ts_newordb_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V8NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_024 of D1
drop tablespace ts_newordb_024;
create regular tablespace ts_newordb_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V9NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_025 of D1
drop tablespace ts_newordb_025;
create regular tablespace ts_newordb_025 pagesize 4K

```

```

        managed by database
        using
        (
            device '/dev/rD1F02V10NORB' 88320
        )
        extentsize 256
        prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_026 of D1
drop tablespace ts_newordb_026;
create regular tablespace ts_newordb_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V11NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_027 of D1
drop tablespace ts_newordb_027;
create regular tablespace ts_newordb_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V12NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_028 of D1
drop tablespace ts_newordb_028;
create regular tablespace ts_newordb_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V13NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_029 of D1
drop tablespace ts_newordb_029;
create regular tablespace ts_newordb_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V14NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_030 of D1
drop tablespace ts_newordb_030;
create regular tablespace ts_newordb_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V15NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_newordb_031 of D1
drop tablespace ts_newordb_031;
create regular tablespace ts_newordb_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_032 of D1
drop tablespace ts_newordb_032;
create regular tablespace ts_newordb_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_033 of D1
drop tablespace ts_newordb_033;
create regular tablespace ts_newordb_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_034 of D1
drop tablespace ts_newordb_034;
create regular tablespace ts_newordb_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_035 of D1
drop tablespace ts_newordb_035;
create regular tablespace ts_newordb_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_newordb_036 of D1
drop tablespace ts_newordb_036;
create regular tablespace ts_newordb_036 pagesize 4K
managed by database
using
(

```

```

        device '/dev/rD1F03V6NORB' 88320
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_037 of D1

drop tablespace ts_neworbd_037;
create regular tablespace ts_neworbd_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V7NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_038 of D1

drop tablespace ts_neworbd_038;
create regular tablespace ts_neworbd_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V8NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_039 of D1

drop tablespace ts_neworbd_039;
create regular tablespace ts_neworbd_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V9NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_040 of D1

drop tablespace ts_neworbd_040;
create regular tablespace ts_neworbd_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V10NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_041 of D1

drop tablespace ts_neworbd_041;
create regular tablespace ts_neworbd_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V11NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_042 of D1

```

```

drop tablespace ts_neworbd_042;
create regular tablespace ts_neworbd_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V12NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_043 of D1

drop tablespace ts_neworbd_043;
create regular tablespace ts_neworbd_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V13NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_044 of D1

drop tablespace ts_neworbd_044;
create regular tablespace ts_neworbd_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V14NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_045 of D1

drop tablespace ts_neworbd_045;
create regular tablespace ts_neworbd_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V15NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_046 of D1

drop tablespace ts_neworbd_046;
create regular tablespace ts_neworbd_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_047 of D1

drop tablespace ts_neworbd_047;
create regular tablespace ts_neworbd_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2NORB' 88320
)
extentsize 256

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_048 of D1

drop tablespace ts_neworbd_048;
create regular tablespace ts_neworbd_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_049 of D1

drop tablespace ts_neworbd_049;
create regular tablespace ts_neworbd_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_050 of D1

drop tablespace ts_neworbd_050;
create regular tablespace ts_neworbd_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_051 of D1

drop tablespace ts_neworbd_051;
create regular tablespace ts_neworbd_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V6NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_052 of D1

drop tablespace ts_neworbd_052;
create regular tablespace ts_neworbd_052 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V7NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_neworbd_053 of D1

drop tablespace ts_neworbd_053;
create regular tablespace ts_neworbd_053 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F04V8NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_054 of D1

drop tablespace ts_newordb_054;
create regular tablespace ts_newordb_054 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V9NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_055 of D1

drop tablespace ts_newordb_055;
create regular tablespace ts_newordb_055 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V10NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_056 of D1

drop tablespace ts_newordb_056;
create regular tablespace ts_newordb_056 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V11NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_057 of D1

drop tablespace ts_newordb_057;
create regular tablespace ts_newordb_057 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V12NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_058 of D1

drop tablespace ts_newordb_058;
create regular tablespace ts_newordb_058 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V13NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_newordb_059 of D1

drop tablespace ts_newordb_059;
create regular tablespace ts_newordb_059 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V14NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_060 of D1

drop tablespace ts_newordb_060;
create regular tablespace ts_newordb_060 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V15NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_061 of D1

drop tablespace ts_newordb_061;
create regular tablespace ts_newordb_061 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_062 of D1

drop tablespace ts_newordb_062;
create regular tablespace ts_newordb_062 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_063 of D1

drop tablespace ts_newordb_063;
create regular tablespace ts_newordb_063 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_064 of D1

drop tablespace ts_newordb_064;
create regular tablespace ts_newordb_064 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4NORB' 88320
)

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_065 of D1

drop tablespace ts_newordb_065;
create regular tablespace ts_newordb_065 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_066 of D1

drop tablespace ts_newordb_066;
create regular tablespace ts_newordb_066 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V6NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_067 of D1

drop tablespace ts_newordb_067;
create regular tablespace ts_newordb_067 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V7NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_068 of D1

drop tablespace ts_newordb_068;
create regular tablespace ts_newordb_068 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V8NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_069 of D1

drop tablespace ts_newordb_069;
create regular tablespace ts_newordb_069 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V9NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_070 of D1

drop tablespace ts_newordb_070;

```

```

create regular tablespace ts_newordb_070 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V10NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_071 of D1

drop tablespace ts_newordb_071;
create regular tablespace ts_newordb_071 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V11NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_072 of D1

drop tablespace ts_newordb_072;
create regular tablespace ts_newordb_072 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V12NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_073 of D1

drop tablespace ts_newordb_073;
create regular tablespace ts_newordb_073 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V13NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_074 of D1

drop tablespace ts_newordb_074;
create regular tablespace ts_newordb_074 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V14NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_075 of D1

drop tablespace ts_newordb_075;
create regular tablespace ts_newordb_075 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V15NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_newordb_076 of D1

drop tablespace ts_newordb_076;
create regular tablespace ts_newordb_076 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V1NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_077 of D1

drop tablespace ts_newordb_077;
create regular tablespace ts_newordb_077 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V2NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_078 of D1

drop tablespace ts_newordb_078;
create regular tablespace ts_newordb_078 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V3NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_079 of D1

drop tablespace ts_newordb_079;
create regular tablespace ts_newordb_079 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V4NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_080 of D1

drop tablespace ts_newordb_080;
create regular tablespace ts_newordb_080 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V5NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_081 of D1

drop tablespace ts_newordb_081;
create regular tablespace ts_newordb_081 pagesize 4K
  managed by database
  using

```

```

  (
    device '/dev/rD1F06V6NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_082 of D1

drop tablespace ts_newordb_082;
create regular tablespace ts_newordb_082 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V7NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_083 of D1

drop tablespace ts_newordb_083;
create regular tablespace ts_newordb_083 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V8NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_084 of D1

drop tablespace ts_newordb_084;
create regular tablespace ts_newordb_084 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V9NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_085 of D1

drop tablespace ts_newordb_085;
create regular tablespace ts_newordb_085 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V10NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_086 of D1

drop tablespace ts_newordb_086;
create regular tablespace ts_newordb_086 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V11NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_087 of D1

```

```

drop tablespace ts_newordb_087;
create regular tablespace ts_newordb_087 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V12NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_088 of D1

drop tablespace ts_newordb_088;
create regular tablespace ts_newordb_088 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V13NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_089 of D1

drop tablespace ts_newordb_089;
create regular tablespace ts_newordb_089 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V14NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_090 of D1

drop tablespace ts_newordb_090;
create regular tablespace ts_newordb_090 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V15NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_091 of D1

drop tablespace ts_newordb_091;
create regular tablespace ts_newordb_091 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V1NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_092 of D1

drop tablespace ts_newordb_092;
create regular tablespace ts_newordb_092 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V2NORB' 88320
  )

```

```

    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_093 of D1

drop tablespace ts_newordb_093;
create regular tablespace ts_newordb_093 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V3NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_094 of D1

drop tablespace ts_newordb_094;
create regular tablespace ts_newordb_094 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V4NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_095 of D1

drop tablespace ts_newordb_095;
create regular tablespace ts_newordb_095 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V5NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_096 of D1

drop tablespace ts_newordb_096;
create regular tablespace ts_newordb_096 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V6NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_097 of D1

drop tablespace ts_newordb_097;
create regular tablespace ts_newordb_097 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V7NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_098 of D1

drop tablespace ts_newordb_098;
create regular tablespace ts_newordb_098 pagesize 4K

```

```

  managed by database
  using
  (
    device '/dev/rD1F07V8NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_099 of D1

drop tablespace ts_newordb_099;
create regular tablespace ts_newordb_099 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V9NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_100 of D1

drop tablespace ts_newordb_100;
create regular tablespace ts_newordb_100 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V10NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_101 of D1

drop tablespace ts_newordb_101;
create regular tablespace ts_newordb_101 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V11NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_102 of D1

drop tablespace ts_newordb_102;
create regular tablespace ts_newordb_102 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V12NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_103 of D1

drop tablespace ts_newordb_103;
create regular tablespace ts_newordb_103 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V13NORB' 88320
  )
  extentsize 256
  prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_newordb_104 of D1

drop tablespace ts_newordb_104;
create regular tablespace ts_newordb_104 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V14NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_105 of D1

drop tablespace ts_newordb_105;
create regular tablespace ts_newordb_105 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V15NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_106 of D1

drop tablespace ts_newordb_106;
create regular tablespace ts_newordb_106 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_107 of D1

drop tablespace ts_newordb_107;
create regular tablespace ts_newordb_107 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_108 of D1

drop tablespace ts_newordb_108;
create regular tablespace ts_newordb_108 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_109 of D1

drop tablespace ts_newordb_109;
create regular tablespace ts_newordb_109 pagesize 4K
managed by database
using
(

```

```

    device '/dev/rD1F08V4NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_110 of D1

drop tablespace ts_newordb_110;
create regular tablespace ts_newordb_110 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_111 of D1

drop tablespace ts_newordb_111;
create regular tablespace ts_newordb_111 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V6NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_112 of D1

drop tablespace ts_newordb_112;
create regular tablespace ts_newordb_112 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V7NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_113 of D1

drop tablespace ts_newordb_113;
create regular tablespace ts_newordb_113 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V8NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_114 of D1

drop tablespace ts_newordb_114;
create regular tablespace ts_newordb_114 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V9NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_115 of D1

```

```

drop tablespace ts_newordb_115;
create regular tablespace ts_newordb_115 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V10NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_116 of D1

drop tablespace ts_newordb_116;
create regular tablespace ts_newordb_116 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V11NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_117 of D1

drop tablespace ts_newordb_117;
create regular tablespace ts_newordb_117 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V12NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_118 of D1

drop tablespace ts_newordb_118;
create regular tablespace ts_newordb_118 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V13NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_119 of D1

drop tablespace ts_newordb_119;
create regular tablespace ts_newordb_119 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V14NORB' 88320
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_newordb_120 of D1

drop tablespace ts_newordb_120;
create regular tablespace ts_newordb_120 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V15NORB' 88320
)
extentsize 256

```

```

prefetchsize 4096;
commit;

connect reset;

ts/crts_order.ddl

connect to tpcc;
-- now creating TS for ts_order_001 of D1

drop tablespace ts_order_001;
create regular tablespace ts_order_001 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_002 of D1

drop tablespace ts_order_002;
create regular tablespace ts_order_002 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_003 of D1

drop tablespace ts_order_003;
create regular tablespace ts_order_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_004 of D1

drop tablespace ts_order_004;
create regular tablespace ts_order_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_005 of D1

drop tablespace ts_order_005;
create regular tablespace ts_order_005 pagesize 8K
managed by database
using

```

```

(
    device '/dev/rD1F01V5ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_006 of D1

drop tablespace ts_order_006;
create regular tablespace ts_order_006 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V6ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_007 of D1

drop tablespace ts_order_007;
create regular tablespace ts_order_007 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V7ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_008 of D1

drop tablespace ts_order_008;
create regular tablespace ts_order_008 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V8ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_009 of D1

drop tablespace ts_order_009;
create regular tablespace ts_order_009 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V9ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_010 of D1

drop tablespace ts_order_010;
create regular tablespace ts_order_010 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V10ORD' 172544
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_011 of D1

drop tablespace ts_order_011;
create regular tablespace ts_order_011 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V11ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_012 of D1

drop tablespace ts_order_012;
create regular tablespace ts_order_012 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V12ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_013 of D1

drop tablespace ts_order_013;
create regular tablespace ts_order_013 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V13ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_014 of D1

drop tablespace ts_order_014;
create regular tablespace ts_order_014 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V14ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_015 of D1

drop tablespace ts_order_015;
create regular tablespace ts_order_015 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V15ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;
-- now creating TS for ts_order_016 of D1
drop tablespace ts_order_016;
create regular tablespace ts_order_016 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V10RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_017 of D1
drop tablespace ts_order_017;
create regular tablespace ts_order_017 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V20RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_018 of D1
drop tablespace ts_order_018;
create regular tablespace ts_order_018 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V30RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_019 of D1
drop tablespace ts_order_019;
create regular tablespace ts_order_019 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V40RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_020 of D1
drop tablespace ts_order_020;
create regular tablespace ts_order_020 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V50RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_021 of D1

```

```

drop tablespace ts_order_021;
create regular tablespace ts_order_021 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V60RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_022 of D1
drop tablespace ts_order_022;
create regular tablespace ts_order_022 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V70RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_023 of D1
drop tablespace ts_order_023;
create regular tablespace ts_order_023 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V80RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_024 of D1
drop tablespace ts_order_024;
create regular tablespace ts_order_024 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V90RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_025 of D1
drop tablespace ts_order_025;
create regular tablespace ts_order_025 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V100RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_026 of D1
drop tablespace ts_order_026;
create regular tablespace ts_order_026 pagesize 8K

```

```

managed by database
using
(
    device '/dev/rD1F02V110RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_027 of D1
drop tablespace ts_order_027;
create regular tablespace ts_order_027 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V120RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_028 of D1
drop tablespace ts_order_028;
create regular tablespace ts_order_028 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V130RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_029 of D1
drop tablespace ts_order_029;
create regular tablespace ts_order_029 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V140RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_030 of D1
drop tablespace ts_order_030;
create regular tablespace ts_order_030 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V150RD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_031 of D1
drop tablespace ts_order_031;
create regular tablespace ts_order_031 pagesize 8K
managed by database
using
(

```



```

        device '/dev/rD1F03V1ORD' 172544
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_032 of D1

drop tablespace ts_order_032;
create regular tablespace ts_order_032 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V2ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_033 of D1

drop tablespace ts_order_033;
create regular tablespace ts_order_033 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V3ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_034 of D1

drop tablespace ts_order_034;
create regular tablespace ts_order_034 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V4ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_035 of D1

drop tablespace ts_order_035;
create regular tablespace ts_order_035 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V5ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_036 of D1

drop tablespace ts_order_036;
create regular tablespace ts_order_036 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V6ORD' 172544
)
extentsize 256

```

```

    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_037 of D1

drop tablespace ts_order_037;
create regular tablespace ts_order_037 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V7ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_038 of D1

drop tablespace ts_order_038;
create regular tablespace ts_order_038 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V8ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_039 of D1

drop tablespace ts_order_039;
create regular tablespace ts_order_039 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V9ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_040 of D1

drop tablespace ts_order_040;
create regular tablespace ts_order_040 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V10ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_041 of D1

drop tablespace ts_order_041;
create regular tablespace ts_order_041 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V11ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_order_042 of D1

drop tablespace ts_order_042;
create regular tablespace ts_order_042 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V12ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_043 of D1

drop tablespace ts_order_043;
create regular tablespace ts_order_043 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V13ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_044 of D1

drop tablespace ts_order_044;
create regular tablespace ts_order_044 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V14ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_045 of D1

drop tablespace ts_order_045;
create regular tablespace ts_order_045 pagesize 8K
managed by database
using
(
    device '/dev/rD1F03V15ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_046 of D1

drop tablespace ts_order_046;
create regular tablespace ts_order_046 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V1ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_047 of D1

```

```

drop tablespace ts_order_047;
create regular tablespace ts_order_047 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V2ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_048 of D1

drop tablespace ts_order_048;
create regular tablespace ts_order_048 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V3ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_049 of D1

drop tablespace ts_order_049;
create regular tablespace ts_order_049 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V4ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_050 of D1

drop tablespace ts_order_050;
create regular tablespace ts_order_050 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V5ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_051 of D1

drop tablespace ts_order_051;
create regular tablespace ts_order_051 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V6ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_052 of D1

drop tablespace ts_order_052;
create regular tablespace ts_order_052 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F04V7ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_053 of D1

drop tablespace ts_order_053;
create regular tablespace ts_order_053 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V8ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_054 of D1

drop tablespace ts_order_054;
create regular tablespace ts_order_054 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V9ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_055 of D1

drop tablespace ts_order_055;
create regular tablespace ts_order_055 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V10ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_056 of D1

drop tablespace ts_order_056;
create regular tablespace ts_order_056 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V11ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_057 of D1

drop tablespace ts_order_057;
create regular tablespace ts_order_057 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V12ORD' 172544

```

```

)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_058 of D1

drop tablespace ts_order_058;
create regular tablespace ts_order_058 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V13ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_059 of D1

drop tablespace ts_order_059;
create regular tablespace ts_order_059 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V14ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_060 of D1

drop tablespace ts_order_060;
create regular tablespace ts_order_060 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V15ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_061 of D1

drop tablespace ts_order_061;
create regular tablespace ts_order_061 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V1ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_062 of D1

drop tablespace ts_order_062;
create regular tablespace ts_order_062 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V2ORD' 172544
)
extentsize 256
prefetchsize 4096

```

```

        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_063 of D1
drop tablespace ts_order_063;
create regular tablespace ts_order_063 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V3ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_064 of D1
drop tablespace ts_order_064;
create regular tablespace ts_order_064 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V4ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_065 of D1
drop tablespace ts_order_065;
create regular tablespace ts_order_065 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V5ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_066 of D1
drop tablespace ts_order_066;
create regular tablespace ts_order_066 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V6ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_067 of D1
drop tablespace ts_order_067;
create regular tablespace ts_order_067 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V7ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_order_068 of D1
drop tablespace ts_order_068;
create regular tablespace ts_order_068 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V8ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_069 of D1
drop tablespace ts_order_069;
create regular tablespace ts_order_069 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V9ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_070 of D1
drop tablespace ts_order_070;
create regular tablespace ts_order_070 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V10ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_071 of D1
drop tablespace ts_order_071;
create regular tablespace ts_order_071 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V11ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_072 of D1
drop tablespace ts_order_072;
create regular tablespace ts_order_072 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V12ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_073 of D1
drop tablespace ts_order_073;

```

```

create regular tablespace ts_order_073 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V13ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_074 of D1
drop tablespace ts_order_074;
create regular tablespace ts_order_074 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V14ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_075 of D1
drop tablespace ts_order_075;
create regular tablespace ts_order_075 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V15ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_076 of D1
drop tablespace ts_order_076;
create regular tablespace ts_order_076 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V1ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_077 of D1
drop tablespace ts_order_077;
create regular tablespace ts_order_077 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V2ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_078 of D1
drop tablespace ts_order_078;
create regular tablespace ts_order_078 pagesize 8K
managed by database
using

```

```

(
  device '/dev/rD1F06V3ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_079 of D1

drop tablespace ts_order_079;
create regular tablespace ts_order_079 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V4ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_080 of D1

drop tablespace ts_order_080;
create regular tablespace ts_order_080 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V5ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_081 of D1

drop tablespace ts_order_081;
create regular tablespace ts_order_081 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V6ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_082 of D1

drop tablespace ts_order_082;
create regular tablespace ts_order_082 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V7ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_083 of D1

drop tablespace ts_order_083;
create regular tablespace ts_order_083 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V8ORD' 172544
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_084 of D1

drop tablespace ts_order_084;
create regular tablespace ts_order_084 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V9ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_085 of D1

drop tablespace ts_order_085;
create regular tablespace ts_order_085 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V10ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_086 of D1

drop tablespace ts_order_086;
create regular tablespace ts_order_086 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V11ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_087 of D1

drop tablespace ts_order_087;
create regular tablespace ts_order_087 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V12ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_088 of D1

drop tablespace ts_order_088;
create regular tablespace ts_order_088 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V13ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for ts_order_089 of D1

drop tablespace ts_order_089;
create regular tablespace ts_order_089 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V14ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_090 of D1

drop tablespace ts_order_090;
create regular tablespace ts_order_090 pagesize 8K
managed by database
using
(
  device '/dev/rD1F06V15ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_091 of D1

drop tablespace ts_order_091;
create regular tablespace ts_order_091 pagesize 8K
managed by database
using
(
  device '/dev/rD1F07V1ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_092 of D1

drop tablespace ts_order_092;
create regular tablespace ts_order_092 pagesize 8K
managed by database
using
(
  device '/dev/rD1F07V2ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_093 of D1

drop tablespace ts_order_093;
create regular tablespace ts_order_093 pagesize 8K
managed by database
using
(
  device '/dev/rD1F07V3ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_094 of D1

```

```

drop tablespace ts_order_094;
create regular tablespace ts_order_094 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V4ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_095 of D1

drop tablespace ts_order_095;
create regular tablespace ts_order_095 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V5ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_096 of D1

drop tablespace ts_order_096;
create regular tablespace ts_order_096 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V6ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_097 of D1

drop tablespace ts_order_097;
create regular tablespace ts_order_097 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V7ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_098 of D1

drop tablespace ts_order_098;
create regular tablespace ts_order_098 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V8ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_099 of D1

drop tablespace ts_order_099;
create regular tablespace ts_order_099 pagesize 8K

```

```

  managed by database
  using
  (
    device '/dev/rD1F07V9ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_100 of D1

drop tablespace ts_order_100;
create regular tablespace ts_order_100 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V10ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_101 of D1

drop tablespace ts_order_101;
create regular tablespace ts_order_101 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V11ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_102 of D1

drop tablespace ts_order_102;
create regular tablespace ts_order_102 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V12ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_103 of D1

drop tablespace ts_order_103;
create regular tablespace ts_order_103 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V13ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_104 of D1

drop tablespace ts_order_104;
create regular tablespace ts_order_104 pagesize 8K
  managed by database
  using
  (

```

```

    device '/dev/rD1F07V14ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_105 of D1

drop tablespace ts_order_105;
create regular tablespace ts_order_105 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V15ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_106 of D1

drop tablespace ts_order_106;
create regular tablespace ts_order_106 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V1ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_107 of D1

drop tablespace ts_order_107;
create regular tablespace ts_order_107 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V2ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_108 of D1

drop tablespace ts_order_108;
create regular tablespace ts_order_108 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V3ORD' 172544
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_109 of D1

drop tablespace ts_order_109;
create regular tablespace ts_order_109 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V4ORD' 172544
  )
  extentsize 256

```

```

        prefetchsize 4096
        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_110 of D1

drop tablespace ts_order_110;
create regular tablespace ts_order_110 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V5ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_111 of D1

drop tablespace ts_order_111;
create regular tablespace ts_order_111 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V6ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_112 of D1

drop tablespace ts_order_112;
create regular tablespace ts_order_112 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V7ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_113 of D1

drop tablespace ts_order_113;
create regular tablespace ts_order_113 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V8ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_114 of D1

drop tablespace ts_order_114;
create regular tablespace ts_order_114 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V9ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_order_115 of D1

drop tablespace ts_order_115;
create regular tablespace ts_order_115 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V10ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_116 of D1

drop tablespace ts_order_116;
create regular tablespace ts_order_116 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V11ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_117 of D1

drop tablespace ts_order_117;
create regular tablespace ts_order_117 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V12ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_118 of D1

drop tablespace ts_order_118;
create regular tablespace ts_order_118 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V13ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_119 of D1

drop tablespace ts_order_119;
create regular tablespace ts_order_119 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V14ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_120 of D1

```

```

drop tablespace ts_order_120;
create regular tablespace ts_order_120 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V15ORD' 172544
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

### ts/crts\_orderline.ddl

```

connect to tpcc;
-- now creating TS for ts_orderline_001 of D1

drop tablespace ts_orderline_001;
create regular tablespace ts_orderline_001 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_002 of D1

drop tablespace ts_orderline_002;
create regular tablespace ts_orderline_002 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_003 of D1

drop tablespace ts_orderline_003;
create regular tablespace ts_orderline_003 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_004 of D1

drop tablespace ts_orderline_004;
create regular tablespace ts_orderline_004 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V4ORL' 4486912
)
extentsize 256
prefetchsize 4096

```

```

        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_005 of D1
drop tablespace ts_orderline_005;
create regular tablespace ts_orderline_005 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V5ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_006 of D1
drop tablespace ts_orderline_006;
create regular tablespace ts_orderline_006 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V6ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_007 of D1
drop tablespace ts_orderline_007;
create regular tablespace ts_orderline_007 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V7ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_008 of D1
drop tablespace ts_orderline_008;
create regular tablespace ts_orderline_008 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V8ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_009 of D1
drop tablespace ts_orderline_009;
create regular tablespace ts_orderline_009 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V9ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_orderline_010 of D1
drop tablespace ts_orderline_010;
create regular tablespace ts_orderline_010 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V10ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_011 of D1
drop tablespace ts_orderline_011;
create regular tablespace ts_orderline_011 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V11ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_012 of D1
drop tablespace ts_orderline_012;
create regular tablespace ts_orderline_012 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V12ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_013 of D1
drop tablespace ts_orderline_013;
create regular tablespace ts_orderline_013 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V13ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_014 of D1
drop tablespace ts_orderline_014;
create regular tablespace ts_orderline_014 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V14ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_015 of D1
drop tablespace ts_orderline_015;

```

```

create regular tablespace ts_orderline_015 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V15ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_016 of D1
drop tablespace ts_orderline_016;
create regular tablespace ts_orderline_016 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V1ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_017 of D1
drop tablespace ts_orderline_017;
create regular tablespace ts_orderline_017 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V2ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_018 of D1
drop tablespace ts_orderline_018;
create regular tablespace ts_orderline_018 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V3ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_019 of D1
drop tablespace ts_orderline_019;
create regular tablespace ts_orderline_019 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V4ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_020 of D1
drop tablespace ts_orderline_020;
create regular tablespace ts_orderline_020 pagesize 8K
managed by database
using

```

```

(
  device '/dev/rD1F02V5ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_021 of D1

drop tablespace ts_orderline_021;
create regular tablespace ts_orderline_021 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V6ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_022 of D1

drop tablespace ts_orderline_022;
create regular tablespace ts_orderline_022 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V7ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_023 of D1

drop tablespace ts_orderline_023;
create regular tablespace ts_orderline_023 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V8ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_024 of D1

drop tablespace ts_orderline_024;
create regular tablespace ts_orderline_024 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V9ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_025 of D1

drop tablespace ts_orderline_025;
create regular tablespace ts_orderline_025 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V10ORL' 4486912
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_026 of D1

drop tablespace ts_orderline_026;
create regular tablespace ts_orderline_026 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V11ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_027 of D1

drop tablespace ts_orderline_027;
create regular tablespace ts_orderline_027 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V12ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_028 of D1

drop tablespace ts_orderline_028;
create regular tablespace ts_orderline_028 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V13ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_029 of D1

drop tablespace ts_orderline_029;
create regular tablespace ts_orderline_029 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V14ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_030 of D1

drop tablespace ts_orderline_030;
create regular tablespace ts_orderline_030 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V15ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for ts_orderline_031 of D1

drop tablespace ts_orderline_031;
create regular tablespace ts_orderline_031 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V1ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_032 of D1

drop tablespace ts_orderline_032;
create regular tablespace ts_orderline_032 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V2ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_033 of D1

drop tablespace ts_orderline_033;
create regular tablespace ts_orderline_033 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V3ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_034 of D1

drop tablespace ts_orderline_034;
create regular tablespace ts_orderline_034 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V4ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_035 of D1

drop tablespace ts_orderline_035;
create regular tablespace ts_orderline_035 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V5ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_036 of D1

```



```

drop tablespace ts_orderline_036;
create regular tablespace ts_orderline_036 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V6ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_037 of D1

drop tablespace ts_orderline_037;
create regular tablespace ts_orderline_037 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V7ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_038 of D1

drop tablespace ts_orderline_038;
create regular tablespace ts_orderline_038 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V8ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_039 of D1

drop tablespace ts_orderline_039;
create regular tablespace ts_orderline_039 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V9ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_040 of D1

drop tablespace ts_orderline_040;
create regular tablespace ts_orderline_040 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V10ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_041 of D1

drop tablespace ts_orderline_041;
create regular tablespace ts_orderline_041 pagesize 8K

```

```

  managed by database
  using
  (
    device '/dev/rD1F03V11ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_042 of D1

drop tablespace ts_orderline_042;
create regular tablespace ts_orderline_042 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V12ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_043 of D1

drop tablespace ts_orderline_043;
create regular tablespace ts_orderline_043 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V13ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_044 of D1

drop tablespace ts_orderline_044;
create regular tablespace ts_orderline_044 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V14ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_045 of D1

drop tablespace ts_orderline_045;
create regular tablespace ts_orderline_045 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V15ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_046 of D1

drop tablespace ts_orderline_046;
create regular tablespace ts_orderline_046 pagesize 8K
  managed by database
  using
  (

```

```

    device '/dev/rD1F04V1ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_047 of D1

drop tablespace ts_orderline_047;
create regular tablespace ts_orderline_047 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V2ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_048 of D1

drop tablespace ts_orderline_048;
create regular tablespace ts_orderline_048 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V3ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_049 of D1

drop tablespace ts_orderline_049;
create regular tablespace ts_orderline_049 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V4ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_050 of D1

drop tablespace ts_orderline_050;
create regular tablespace ts_orderline_050 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V5ORL' 4486912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_051 of D1

drop tablespace ts_orderline_051;
create regular tablespace ts_orderline_051 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V6ORL' 4486912
  )
  extentsize 256

```

```

        prefetchsize 4096
        bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_052 of D1

drop tablespace ts_orderline_052;
create regular tablespace ts_orderline_052 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V7ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_053 of D1

drop tablespace ts_orderline_053;
create regular tablespace ts_orderline_053 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V8ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_054 of D1

drop tablespace ts_orderline_054;
create regular tablespace ts_orderline_054 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V9ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_055 of D1

drop tablespace ts_orderline_055;
create regular tablespace ts_orderline_055 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V10ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_056 of D1

drop tablespace ts_orderline_056;
create regular tablespace ts_orderline_056 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V11ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_orderline_057 of D1

drop tablespace ts_orderline_057;
create regular tablespace ts_orderline_057 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V12ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_058 of D1

drop tablespace ts_orderline_058;
create regular tablespace ts_orderline_058 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V13ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_059 of D1

drop tablespace ts_orderline_059;
create regular tablespace ts_orderline_059 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V14ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_060 of D1

drop tablespace ts_orderline_060;
create regular tablespace ts_orderline_060 pagesize 8K
managed by database
using
(
    device '/dev/rD1F04V15ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_061 of D1

drop tablespace ts_orderline_061;
create regular tablespace ts_orderline_061 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V1ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_062 of D1

```

```

drop tablespace ts_orderline_062;
create regular tablespace ts_orderline_062 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V2ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_063 of D1

drop tablespace ts_orderline_063;
create regular tablespace ts_orderline_063 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V3ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_064 of D1

drop tablespace ts_orderline_064;
create regular tablespace ts_orderline_064 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V4ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_065 of D1

drop tablespace ts_orderline_065;
create regular tablespace ts_orderline_065 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V5ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_066 of D1

drop tablespace ts_orderline_066;
create regular tablespace ts_orderline_066 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V6ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_067 of D1

drop tablespace ts_orderline_067;
create regular tablespace ts_orderline_067 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F05V7ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_068 of D1

drop tablespace ts_orderline_068;
create regular tablespace ts_orderline_068 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V8ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_069 of D1

drop tablespace ts_orderline_069;
create regular tablespace ts_orderline_069 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V9ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_070 of D1

drop tablespace ts_orderline_070;
create regular tablespace ts_orderline_070 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V10ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_071 of D1

drop tablespace ts_orderline_071;
create regular tablespace ts_orderline_071 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V11ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_072 of D1

drop tablespace ts_orderline_072;
create regular tablespace ts_orderline_072 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V12ORL' 4486912

```

```

)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_073 of D1

drop tablespace ts_orderline_073;
create regular tablespace ts_orderline_073 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V13ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_074 of D1

drop tablespace ts_orderline_074;
create regular tablespace ts_orderline_074 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V14ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_075 of D1

drop tablespace ts_orderline_075;
create regular tablespace ts_orderline_075 pagesize 8K
managed by database
using
(
    device '/dev/rD1F05V15ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_076 of D1

drop tablespace ts_orderline_076;
create regular tablespace ts_orderline_076 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V1ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_077 of D1

drop tablespace ts_orderline_077;
create regular tablespace ts_orderline_077 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V2ORL' 4486912
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_078 of D1

drop tablespace ts_orderline_078;
create regular tablespace ts_orderline_078 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V3ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_079 of D1

drop tablespace ts_orderline_079;
create regular tablespace ts_orderline_079 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V4ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_080 of D1

drop tablespace ts_orderline_080;
create regular tablespace ts_orderline_080 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V5ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_081 of D1

drop tablespace ts_orderline_081;
create regular tablespace ts_orderline_081 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V6ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_082 of D1

drop tablespace ts_orderline_082;
create regular tablespace ts_orderline_082 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V7ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_orderline_083 of D1
drop tablespace ts_orderline_083;
create regular tablespace ts_orderline_083 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V8ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_084 of D1
drop tablespace ts_orderline_084;
create regular tablespace ts_orderline_084 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V9ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_085 of D1
drop tablespace ts_orderline_085;
create regular tablespace ts_orderline_085 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V10ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_086 of D1
drop tablespace ts_orderline_086;
create regular tablespace ts_orderline_086 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V11ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_087 of D1
drop tablespace ts_orderline_087;
create regular tablespace ts_orderline_087 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V12ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_088 of D1
drop tablespace ts_orderline_088;

```

```

create regular tablespace ts_orderline_088 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V13ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_089 of D1
drop tablespace ts_orderline_089;
create regular tablespace ts_orderline_089 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V14ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_090 of D1
drop tablespace ts_orderline_090;
create regular tablespace ts_orderline_090 pagesize 8K
managed by database
using
(
    device '/dev/rD1F06V15ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_091 of D1
drop tablespace ts_orderline_091;
create regular tablespace ts_orderline_091 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V1ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_092 of D1
drop tablespace ts_orderline_092;
create regular tablespace ts_orderline_092 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V2ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_093 of D1
drop tablespace ts_orderline_093;
create regular tablespace ts_orderline_093 pagesize 8K
managed by database
using

```

```

(
    device '/dev/rD1F07V3ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_094 of D1
drop tablespace ts_orderline_094;
create regular tablespace ts_orderline_094 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V4ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_095 of D1
drop tablespace ts_orderline_095;
create regular tablespace ts_orderline_095 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V5ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_096 of D1
drop tablespace ts_orderline_096;
create regular tablespace ts_orderline_096 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V6ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_097 of D1
drop tablespace ts_orderline_097;
create regular tablespace ts_orderline_097 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V7ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_098 of D1
drop tablespace ts_orderline_098;
create regular tablespace ts_orderline_098 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V8ORL' 4486912
)

```

```

        extentsize 256
        prefetchsize 4096
        bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_099 of D1

drop tablespace ts_orderline_099;
create regular tablespace ts_orderline_099 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V9ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_100 of D1

drop tablespace ts_orderline_100;
create regular tablespace ts_orderline_100 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V10ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_101 of D1

drop tablespace ts_orderline_101;
create regular tablespace ts_orderline_101 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V11ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_102 of D1

drop tablespace ts_orderline_102;
create regular tablespace ts_orderline_102 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V12ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_103 of D1

drop tablespace ts_orderline_103;
create regular tablespace ts_orderline_103 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V13ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for ts_orderline_104 of D1

drop tablespace ts_orderline_104;
create regular tablespace ts_orderline_104 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V14ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_105 of D1

drop tablespace ts_orderline_105;
create regular tablespace ts_orderline_105 pagesize 8K
managed by database
using
(
    device '/dev/rD1F07V15ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_106 of D1

drop tablespace ts_orderline_106;
create regular tablespace ts_orderline_106 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V1ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_107 of D1

drop tablespace ts_orderline_107;
create regular tablespace ts_orderline_107 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V2ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_108 of D1

drop tablespace ts_orderline_108;
create regular tablespace ts_orderline_108 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V3ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_109 of D1

```

```

drop tablespace ts_orderline_109;
create regular tablespace ts_orderline_109 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V4ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_110 of D1

drop tablespace ts_orderline_110;
create regular tablespace ts_orderline_110 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V5ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_111 of D1

drop tablespace ts_orderline_111;
create regular tablespace ts_orderline_111 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V6ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_112 of D1

drop tablespace ts_orderline_112;
create regular tablespace ts_orderline_112 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V7ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_113 of D1

drop tablespace ts_orderline_113;
create regular tablespace ts_orderline_113 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V8ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_114 of D1

drop tablespace ts_orderline_114;
create regular tablespace ts_orderline_114 pagesize 8K

```

```

managed by database
using
(
    device '/dev/rD1F08V9ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_115 of D1

drop tablespace ts_orderline_115;
create regular tablespace ts_orderline_115 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V10ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_116 of D1

drop tablespace ts_orderline_116;
create regular tablespace ts_orderline_116 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V11ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_117 of D1

drop tablespace ts_orderline_117;
create regular tablespace ts_orderline_117 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V12ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_118 of D1

drop tablespace ts_orderline_118;
create regular tablespace ts_orderline_118 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V13ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_119 of D1

drop tablespace ts_orderline_119;
create regular tablespace ts_orderline_119 pagesize 8K
managed by database
using
(

```

```

    device '/dev/rD1F08V14ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_120 of D1

drop tablespace ts_orderline_120;
create regular tablespace ts_orderline_120 pagesize 8K
managed by database
using
(
    device '/dev/rD1F08V15ORL' 4486912
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

connect reset;

ts/crts_stock.ddl

connect to tpc;
-- now creating TS for ts_stock_001 of D1

drop tablespace ts_stock_001;
create regular tablespace ts_stock_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_002 of D1

drop tablespace ts_stock_002;
create regular tablespace ts_stock_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_003 of D1

drop tablespace ts_stock_003;
create regular tablespace ts_stock_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_004 of D1

drop tablespace ts_stock_004;
create regular tablespace ts_stock_004 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F01V4STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_005 of D1

drop tablespace ts_stock_005;
create regular tablespace ts_stock_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_006 of D1

drop tablespace ts_stock_006;
create regular tablespace ts_stock_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V6STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_007 of D1

drop tablespace ts_stock_007;
create regular tablespace ts_stock_007 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V7STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_008 of D1

drop tablespace ts_stock_008;
create regular tablespace ts_stock_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V8STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_009 of D1

drop tablespace ts_stock_009;
create regular tablespace ts_stock_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V9STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_stock_010 of D1

drop tablespace ts_stock_010;
create regular tablespace ts_stock_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V10STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_011 of D1

drop tablespace ts_stock_011;
create regular tablespace ts_stock_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V11STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_012 of D1

drop tablespace ts_stock_012;
create regular tablespace ts_stock_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V12STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_013 of D1

drop tablespace ts_stock_013;
create regular tablespace ts_stock_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V13STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_014 of D1

drop tablespace ts_stock_014;
create regular tablespace ts_stock_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V14STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_015 of D1

drop tablespace ts_stock_015;
create regular tablespace ts_stock_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V15STK' 7587840

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_016 of D1

drop tablespace ts_stock_016;
create regular tablespace ts_stock_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_017 of D1

drop tablespace ts_stock_017;
create regular tablespace ts_stock_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_018 of D1

drop tablespace ts_stock_018;
create regular tablespace ts_stock_018 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V3STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_019 of D1

drop tablespace ts_stock_019;
create regular tablespace ts_stock_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_020 of D1

drop tablespace ts_stock_020;
create regular tablespace ts_stock_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_021 of D1

drop tablespace ts_stock_021;

```

```

create regular tablespace ts_stock_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V6STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_022 of D1

drop tablespace ts_stock_022;
create regular tablespace ts_stock_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V7STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_023 of D1

drop tablespace ts_stock_023;
create regular tablespace ts_stock_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V8STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_024 of D1

drop tablespace ts_stock_024;
create regular tablespace ts_stock_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V9STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_025 of D1

drop tablespace ts_stock_025;
create regular tablespace ts_stock_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V10STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_026 of D1

drop tablespace ts_stock_026;
create regular tablespace ts_stock_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V11STK' 7587840
)
extentsize 256
prefetchsize 4096;

```

```

commit;
-- now creating TS for ts_stock_027 of D1
drop tablespace ts_stock_027;
create regular tablespace ts_stock_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V12STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_028 of D1
drop tablespace ts_stock_028;
create regular tablespace ts_stock_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V13STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_029 of D1
drop tablespace ts_stock_029;
create regular tablespace ts_stock_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V14STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_030 of D1
drop tablespace ts_stock_030;
create regular tablespace ts_stock_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V15STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_031 of D1
drop tablespace ts_stock_031;
create regular tablespace ts_stock_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_032 of D1
drop tablespace ts_stock_032;
create regular tablespace ts_stock_032 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F03V2STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_033 of D1
drop tablespace ts_stock_033;
create regular tablespace ts_stock_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_034 of D1
drop tablespace ts_stock_034;
create regular tablespace ts_stock_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_035 of D1
drop tablespace ts_stock_035;
create regular tablespace ts_stock_035 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V5STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_036 of D1
drop tablespace ts_stock_036;
create regular tablespace ts_stock_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V6STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_037 of D1
drop tablespace ts_stock_037;
create regular tablespace ts_stock_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V7STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_038 of D1

```

```

drop tablespace ts_stock_038;
create regular tablespace ts_stock_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V8STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_039 of D1
drop tablespace ts_stock_039;
create regular tablespace ts_stock_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V9STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_040 of D1
drop tablespace ts_stock_040;
create regular tablespace ts_stock_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V10STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_041 of D1
drop tablespace ts_stock_041;
create regular tablespace ts_stock_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V11STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_042 of D1
drop tablespace ts_stock_042;
create regular tablespace ts_stock_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V12STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_043 of D1
drop tablespace ts_stock_043;
create regular tablespace ts_stock_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V13STK' 7587840
)

```



```

        extentsize 256
        prefetchsize 4096;
commit;
-- now creating TS for ts_stock_044 of D1
drop tablespace ts_stock_044;
create regular tablespace ts_stock_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V14STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_045 of D1
drop tablespace ts_stock_045;
create regular tablespace ts_stock_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V15STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_046 of D1
drop tablespace ts_stock_046;
create regular tablespace ts_stock_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_047 of D1
drop tablespace ts_stock_047;
create regular tablespace ts_stock_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_048 of D1
drop tablespace ts_stock_048;
create regular tablespace ts_stock_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_049 of D1
drop tablespace ts_stock_049;
create regular tablespace ts_stock_049 pagesize 4K

```

```

        managed by database
        using
        (
            device '/dev/rD1F04V4STK' 7587840
        )
        extentsize 256
        prefetchsize 4096;
commit;
-- now creating TS for ts_stock_050 of D1
drop tablespace ts_stock_050;
create regular tablespace ts_stock_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_051 of D1
drop tablespace ts_stock_051;
create regular tablespace ts_stock_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V6STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_052 of D1
drop tablespace ts_stock_052;
create regular tablespace ts_stock_052 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V7STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_053 of D1
drop tablespace ts_stock_053;
create regular tablespace ts_stock_053 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V8STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_054 of D1
drop tablespace ts_stock_054;
create regular tablespace ts_stock_054 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V9STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_stock_055 of D1
drop tablespace ts_stock_055;
create regular tablespace ts_stock_055 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V10STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_056 of D1
drop tablespace ts_stock_056;
create regular tablespace ts_stock_056 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V11STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_057 of D1
drop tablespace ts_stock_057;
create regular tablespace ts_stock_057 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V12STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_058 of D1
drop tablespace ts_stock_058;
create regular tablespace ts_stock_058 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V13STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_059 of D1
drop tablespace ts_stock_059;
create regular tablespace ts_stock_059 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V14STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_060 of D1
drop tablespace ts_stock_060;
create regular tablespace ts_stock_060 pagesize 4K
managed by database
using
(

```

```

        device '/dev/rD1F04V15STK' 7587840
    )
    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_061 of D1

drop tablespace ts_stock_061;
create regular tablespace ts_stock_061 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_062 of D1

drop tablespace ts_stock_062;
create regular tablespace ts_stock_062 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_063 of D1

drop tablespace ts_stock_063;
create regular tablespace ts_stock_063 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_064 of D1

drop tablespace ts_stock_064;
create regular tablespace ts_stock_064 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_065 of D1

drop tablespace ts_stock_065;
create regular tablespace ts_stock_065 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_066 of D1

```

```

drop tablespace ts_stock_066;
create regular tablespace ts_stock_066 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V6STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_067 of D1

drop tablespace ts_stock_067;
create regular tablespace ts_stock_067 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V7STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_068 of D1

drop tablespace ts_stock_068;
create regular tablespace ts_stock_068 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V8STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_069 of D1

drop tablespace ts_stock_069;
create regular tablespace ts_stock_069 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V9STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_070 of D1

drop tablespace ts_stock_070;
create regular tablespace ts_stock_070 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V10STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_071 of D1

drop tablespace ts_stock_071;
create regular tablespace ts_stock_071 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V11STK' 7587840
)
extentsize 256

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_072 of D1

drop tablespace ts_stock_072;
create regular tablespace ts_stock_072 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V12STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_073 of D1

drop tablespace ts_stock_073;
create regular tablespace ts_stock_073 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V13STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_074 of D1

drop tablespace ts_stock_074;
create regular tablespace ts_stock_074 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V14STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_075 of D1

drop tablespace ts_stock_075;
create regular tablespace ts_stock_075 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V15STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_076 of D1

drop tablespace ts_stock_076;
create regular tablespace ts_stock_076 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_077 of D1

drop tablespace ts_stock_077;
create regular tablespace ts_stock_077 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F06V2STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_078 of D1

drop tablespace ts_stock_078;
create regular tablespace ts_stock_078 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_079 of D1

drop tablespace ts_stock_079;
create regular tablespace ts_stock_079 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_080 of D1

drop tablespace ts_stock_080;
create regular tablespace ts_stock_080 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V5STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_081 of D1

drop tablespace ts_stock_081;
create regular tablespace ts_stock_081 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V6STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_082 of D1

drop tablespace ts_stock_082;
create regular tablespace ts_stock_082 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V7STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_stock_083 of D1

drop tablespace ts_stock_083;
create regular tablespace ts_stock_083 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V8STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_084 of D1

drop tablespace ts_stock_084;
create regular tablespace ts_stock_084 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V9STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_085 of D1

drop tablespace ts_stock_085;
create regular tablespace ts_stock_085 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V10STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_086 of D1

drop tablespace ts_stock_086;
create regular tablespace ts_stock_086 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V11STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_087 of D1

drop tablespace ts_stock_087;
create regular tablespace ts_stock_087 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V12STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_088 of D1

drop tablespace ts_stock_088;
create regular tablespace ts_stock_088 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V13STK' 7587840

```

```

)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_089 of D1

drop tablespace ts_stock_089;
create regular tablespace ts_stock_089 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V14STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_090 of D1

drop tablespace ts_stock_090;
create regular tablespace ts_stock_090 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V15STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_091 of D1

drop tablespace ts_stock_091;
create regular tablespace ts_stock_091 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_092 of D1

drop tablespace ts_stock_092;
create regular tablespace ts_stock_092 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_093 of D1

drop tablespace ts_stock_093;
create regular tablespace ts_stock_093 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_094 of D1

drop tablespace ts_stock_094;

```

```

create regular tablespace ts_stock_094 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V4STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_095 of D1

drop tablespace ts_stock_095;
create regular tablespace ts_stock_095 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V5STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_096 of D1

drop tablespace ts_stock_096;
create regular tablespace ts_stock_096 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V6STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_097 of D1

drop tablespace ts_stock_097;
create regular tablespace ts_stock_097 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V7STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_098 of D1

drop tablespace ts_stock_098;
create regular tablespace ts_stock_098 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V8STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_099 of D1

drop tablespace ts_stock_099;
create regular tablespace ts_stock_099 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V9STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;

```

```

commit;

-- now creating TS for ts_stock_100 of D1

drop tablespace ts_stock_100;
create regular tablespace ts_stock_100 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V10STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_101 of D1

drop tablespace ts_stock_101;
create regular tablespace ts_stock_101 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V11STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_102 of D1

drop tablespace ts_stock_102;
create regular tablespace ts_stock_102 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V12STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_103 of D1

drop tablespace ts_stock_103;
create regular tablespace ts_stock_103 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V13STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_104 of D1

drop tablespace ts_stock_104;
create regular tablespace ts_stock_104 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V14STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_105 of D1

drop tablespace ts_stock_105;
create regular tablespace ts_stock_105 pagesize 4K
  managed by database
  using

```

```

  (
    device '/dev/rD1F07V15STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_106 of D1

drop tablespace ts_stock_106;
create regular tablespace ts_stock_106 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V1STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_107 of D1

drop tablespace ts_stock_107;
create regular tablespace ts_stock_107 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V2STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_108 of D1

drop tablespace ts_stock_108;
create regular tablespace ts_stock_108 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V3STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_109 of D1

drop tablespace ts_stock_109;
create regular tablespace ts_stock_109 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V4STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_110 of D1

drop tablespace ts_stock_110;
create regular tablespace ts_stock_110 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V5STK' 7587840
  )
  extentsize 256
  prefetchsize 4096;
commit;

-- now creating TS for ts_stock_111 of D1

```

```

drop tablespace ts_stock_111;
create regular tablespace ts_stock_111 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V6STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_112 of D1

drop tablespace ts_stock_112;
create regular tablespace ts_stock_112 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V7STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_113 of D1

drop tablespace ts_stock_113;
create regular tablespace ts_stock_113 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V8STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_114 of D1

drop tablespace ts_stock_114;
create regular tablespace ts_stock_114 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V9STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_115 of D1

drop tablespace ts_stock_115;
create regular tablespace ts_stock_115 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V10STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_116 of D1

drop tablespace ts_stock_116;
create regular tablespace ts_stock_116 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V11STK' 7587840
)

```

```

    extentsize 256
    prefetchsize 4096;
commit;

-- now creating TS for ts_stock_117 of D1

drop tablespace ts_stock_117;
create regular tablespace ts_stock_117 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V12STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_118 of D1

drop tablespace ts_stock_118;
create regular tablespace ts_stock_118 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V13STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_119 of D1

drop tablespace ts_stock_119;
create regular tablespace ts_stock_119 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V14STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

-- now creating TS for ts_stock_120 of D1

drop tablespace ts_stock_120;
create regular tablespace ts_stock_120 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V15STK' 7587840
)
extentsize 256
prefetchsize 4096;
commit;

connect reset;

ts/crts ware.ddl

connect to tpc;
-- now creating TS for ts_ware_001 of D1

drop tablespace ts_ware_001;
create regular tablespace ts_ware_001 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V1WARE' 224
)

```

```

    extentsize 32
    prefetchsize 4096;
commit;

-- now creating TS for ts_ware_002 of D1

drop tablespace ts_ware_002;
create regular tablespace ts_ware_002 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_003 of D1

drop tablespace ts_ware_003;
create regular tablespace ts_ware_003 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_004 of D1

drop tablespace ts_ware_004;
create regular tablespace ts_ware_004 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_005 of D1

drop tablespace ts_ware_005;
create regular tablespace ts_ware_005 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V5WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_006 of D1

drop tablespace ts_ware_006;
create regular tablespace ts_ware_006 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_007 of D1

drop tablespace ts_ware_007;
create regular tablespace ts_ware_007 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F01V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_008 of D1

drop tablespace ts_ware_008;
create regular tablespace ts_ware_008 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_009 of D1

drop tablespace ts_ware_009;
create regular tablespace ts_ware_009 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_010 of D1

drop tablespace ts_ware_010;
create regular tablespace ts_ware_010 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_011 of D1

drop tablespace ts_ware_011;
create regular tablespace ts_ware_011 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_012 of D1

drop tablespace ts_ware_012;
create regular tablespace ts_ware_012 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V12WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_ware_013 of D1

drop tablespace ts_ware_013;
create regular tablespace ts_ware_013 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_014 of D1

drop tablespace ts_ware_014;
create regular tablespace ts_ware_014 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V14WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_015 of D1

drop tablespace ts_ware_015;
create regular tablespace ts_ware_015 pagesize 4K
managed by database
using
(
    device '/dev/rD1F01V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_016 of D1

drop tablespace ts_ware_016;
create regular tablespace ts_ware_016 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V1WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_017 of D1

drop tablespace ts_ware_017;
create regular tablespace ts_ware_017 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_018 of D1

drop tablespace ts_ware_018;
create regular tablespace ts_ware_018 pagesize 4K
managed by database
using
(

```

```

    device '/dev/rD1F02V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_019 of D1

drop tablespace ts_ware_019;
create regular tablespace ts_ware_019 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_020 of D1

drop tablespace ts_ware_020;
create regular tablespace ts_ware_020 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V5WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_021 of D1

drop tablespace ts_ware_021;
create regular tablespace ts_ware_021 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_022 of D1

drop tablespace ts_ware_022;
create regular tablespace ts_ware_022 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_023 of D1

drop tablespace ts_ware_023;
create regular tablespace ts_ware_023 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_024 of D1

```

```

drop tablespace ts_ware_024;
create regular tablespace ts_ware_024 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_025 of D1

drop tablespace ts_ware_025;
create regular tablespace ts_ware_025 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_026 of D1

drop tablespace ts_ware_026;
create regular tablespace ts_ware_026 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_027 of D1

drop tablespace ts_ware_027;
create regular tablespace ts_ware_027 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V12WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_028 of D1

drop tablespace ts_ware_028;
create regular tablespace ts_ware_028 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_029 of D1

drop tablespace ts_ware_029;
create regular tablespace ts_ware_029 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V14WARE' 224
)
extentsize 32

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_ware_030 of D1

drop tablespace ts_ware_030;
create regular tablespace ts_ware_030 pagesize 4K
managed by database
using
(
    device '/dev/rD1F02V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_031 of D1

drop tablespace ts_ware_031;
create regular tablespace ts_ware_031 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V1WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_032 of D1

drop tablespace ts_ware_032;
create regular tablespace ts_ware_032 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_033 of D1

drop tablespace ts_ware_033;
create regular tablespace ts_ware_033 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_034 of D1

drop tablespace ts_ware_034;
create regular tablespace ts_ware_034 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_035 of D1

drop tablespace ts_ware_035;
create regular tablespace ts_ware_035 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F03V5WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_036 of D1

drop tablespace ts_ware_036;
create regular tablespace ts_ware_036 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_037 of D1

drop tablespace ts_ware_037;
create regular tablespace ts_ware_037 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_038 of D1

drop tablespace ts_ware_038;
create regular tablespace ts_ware_038 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_039 of D1

drop tablespace ts_ware_039;
create regular tablespace ts_ware_039 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_040 of D1

drop tablespace ts_ware_040;
create regular tablespace ts_ware_040 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_ware_041 of D1
drop tablespace ts_ware_041;
create regular tablespace ts_ware_041 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_042 of D1
drop tablespace ts_ware_042;
create regular tablespace ts_ware_042 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V12WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_043 of D1
drop tablespace ts_ware_043;
create regular tablespace ts_ware_043 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_044 of D1
drop tablespace ts_ware_044;
create regular tablespace ts_ware_044 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V14WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_045 of D1
drop tablespace ts_ware_045;
create regular tablespace ts_ware_045 pagesize 4K
managed by database
using
(
    device '/dev/rD1F03V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_046 of D1
drop tablespace ts_ware_046;
create regular tablespace ts_ware_046 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V1WARE' 224

```

```

)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_047 of D1
drop tablespace ts_ware_047;
create regular tablespace ts_ware_047 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_048 of D1
drop tablespace ts_ware_048;
create regular tablespace ts_ware_048 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_049 of D1
drop tablespace ts_ware_049;
create regular tablespace ts_ware_049 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_050 of D1
drop tablespace ts_ware_050;
create regular tablespace ts_ware_050 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V5WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_051 of D1
drop tablespace ts_ware_051;
create regular tablespace ts_ware_051 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_052 of D1
drop tablespace ts_ware_052;

```

```

create regular tablespace ts_ware_052 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_053 of D1
drop tablespace ts_ware_053;
create regular tablespace ts_ware_053 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_054 of D1
drop tablespace ts_ware_054;
create regular tablespace ts_ware_054 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_055 of D1
drop tablespace ts_ware_055;
create regular tablespace ts_ware_055 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_056 of D1
drop tablespace ts_ware_056;
create regular tablespace ts_ware_056 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_057 of D1
drop tablespace ts_ware_057;
create regular tablespace ts_ware_057 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V12WARE' 224
)
extentsize 32
prefetchsize 4096;

```



```

commit;
-- now creating TS for ts_ware_058 of D1
drop tablespace ts_ware_058;
create regular tablespace ts_ware_058 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_059 of D1
drop tablespace ts_ware_059;
create regular tablespace ts_ware_059 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V14WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_060 of D1
drop tablespace ts_ware_060;
create regular tablespace ts_ware_060 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_061 of D1
drop tablespace ts_ware_061;
create regular tablespace ts_ware_061 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_062 of D1
drop tablespace ts_ware_062;
create regular tablespace ts_ware_062 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_063 of D1
drop tablespace ts_ware_063;
create regular tablespace ts_ware_063 pagesize 4K
managed by database
using

```

```

(
    device '/dev/rD1F05V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_064 of D1
drop tablespace ts_ware_064;
create regular tablespace ts_ware_064 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_065 of D1
drop tablespace ts_ware_065;
create regular tablespace ts_ware_065 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V5WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_066 of D1
drop tablespace ts_ware_066;
create regular tablespace ts_ware_066 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_067 of D1
drop tablespace ts_ware_067;
create regular tablespace ts_ware_067 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_068 of D1
drop tablespace ts_ware_068;
create regular tablespace ts_ware_068 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_069 of D1

```

```

drop tablespace ts_ware_069;
create regular tablespace ts_ware_069 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_070 of D1
drop tablespace ts_ware_070;
create regular tablespace ts_ware_070 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_071 of D1
drop tablespace ts_ware_071;
create regular tablespace ts_ware_071 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_072 of D1
drop tablespace ts_ware_072;
create regular tablespace ts_ware_072 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V12WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_073 of D1
drop tablespace ts_ware_073;
create regular tablespace ts_ware_073 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_074 of D1
drop tablespace ts_ware_074;
create regular tablespace ts_ware_074 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V14WARE' 224
)

```

```

        extentsize 32
        prefetchsize 4096;
commit;
-- now creating TS for ts_ware_075 of D1
drop tablespace ts_ware_075;
create regular tablespace ts_ware_075 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_076 of D1
drop tablespace ts_ware_076;
create regular tablespace ts_ware_076 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_077 of D1
drop tablespace ts_ware_077;
create regular tablespace ts_ware_077 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_078 of D1
drop tablespace ts_ware_078;
create regular tablespace ts_ware_078 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_079 of D1
drop tablespace ts_ware_079;
create regular tablespace ts_ware_079 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_080 of D1
drop tablespace ts_ware_080;
create regular tablespace ts_ware_080 pagesize 4K

```

```

        managed by database
        using
        (
            device '/dev/rD1F06V5WARE' 224
        )
        extentsize 32
        prefetchsize 4096;
commit;
-- now creating TS for ts_ware_081 of D1
drop tablespace ts_ware_081;
create regular tablespace ts_ware_081 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_082 of D1
drop tablespace ts_ware_082;
create regular tablespace ts_ware_082 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_083 of D1
drop tablespace ts_ware_083;
create regular tablespace ts_ware_083 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_084 of D1
drop tablespace ts_ware_084;
create regular tablespace ts_ware_084 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_085 of D1
drop tablespace ts_ware_085;
create regular tablespace ts_ware_085 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_ware_086 of D1
drop tablespace ts_ware_086;
create regular tablespace ts_ware_086 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_087 of D1
drop tablespace ts_ware_087;
create regular tablespace ts_ware_087 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V12WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_088 of D1
drop tablespace ts_ware_088;
create regular tablespace ts_ware_088 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_089 of D1
drop tablespace ts_ware_089;
create regular tablespace ts_ware_089 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V14WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_090 of D1
drop tablespace ts_ware_090;
create regular tablespace ts_ware_090 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;
-- now creating TS for ts_ware_091 of D1
drop tablespace ts_ware_091;
create regular tablespace ts_ware_091 pagesize 4K
managed by database
using
(

```

```

        device '/dev/rD1F07V1WARE' 224
    )
    extentsize 32
    prefetchsize 4096;
commit;

-- now creating TS for ts_ware_092 of D1

drop tablespace ts_ware_092;
create regular tablespace ts_ware_092 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_093 of D1

drop tablespace ts_ware_093;
create regular tablespace ts_ware_093 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_094 of D1

drop tablespace ts_ware_094;
create regular tablespace ts_ware_094 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_095 of D1

drop tablespace ts_ware_095;
create regular tablespace ts_ware_095 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V5WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_096 of D1

drop tablespace ts_ware_096;
create regular tablespace ts_ware_096 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_097 of D1

```

```

drop tablespace ts_ware_097;
create regular tablespace ts_ware_097 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_098 of D1

drop tablespace ts_ware_098;
create regular tablespace ts_ware_098 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_099 of D1

drop tablespace ts_ware_099;
create regular tablespace ts_ware_099 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_100 of D1

drop tablespace ts_ware_100;
create regular tablespace ts_ware_100 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_101 of D1

drop tablespace ts_ware_101;
create regular tablespace ts_ware_101 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_102 of D1

drop tablespace ts_ware_102;
create regular tablespace ts_ware_102 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V12WARE' 224
)
extentsize 32

```

```

    prefetchsize 4096;
commit;

-- now creating TS for ts_ware_103 of D1

drop tablespace ts_ware_103;
create regular tablespace ts_ware_103 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_104 of D1

drop tablespace ts_ware_104;
create regular tablespace ts_ware_104 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V14WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_105 of D1

drop tablespace ts_ware_105;
create regular tablespace ts_ware_105 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_106 of D1

drop tablespace ts_ware_106;
create regular tablespace ts_ware_106 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_107 of D1

drop tablespace ts_ware_107;
create regular tablespace ts_ware_107 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_108 of D1

drop tablespace ts_ware_108;
create regular tablespace ts_ware_108 pagesize 4K
managed by database

```

```

using
(
    device '/dev/rD1F08V3WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_109 of D1

drop tablespace ts_ware_109;
create regular tablespace ts_ware_109 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V4WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_110 of D1

drop tablespace ts_ware_110;
create regular tablespace ts_ware_110 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V5WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_111 of D1

drop tablespace ts_ware_111;
create regular tablespace ts_ware_111 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V6WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_112 of D1

drop tablespace ts_ware_112;
create regular tablespace ts_ware_112 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V7WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_113 of D1

drop tablespace ts_ware_113;
create regular tablespace ts_ware_113 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V8WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_ware_114 of D1

drop tablespace ts_ware_114;
create regular tablespace ts_ware_114 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V9WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_115 of D1

drop tablespace ts_ware_115;
create regular tablespace ts_ware_115 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V10WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_116 of D1

drop tablespace ts_ware_116;
create regular tablespace ts_ware_116 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V11WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_117 of D1

drop tablespace ts_ware_117;
create regular tablespace ts_ware_117 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V12WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_118 of D1

drop tablespace ts_ware_118;
create regular tablespace ts_ware_118 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V13WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_119 of D1

drop tablespace ts_ware_119;
create regular tablespace ts_ware_119 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V14WARE' 224
)

```

```

)
extentsize 32
prefetchsize 4096;
commit;

-- now creating TS for ts_ware_120 of D1

drop tablespace ts_ware_120;
create regular tablespace ts_ware_120 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V15WARE' 224
)
extentsize 32
prefetchsize 4096;
commit;

connect reset;

```

## C.2 Data Generation 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 - AIX 64-bit
#
#
# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g", "-DDEBUG" "-g -DDEBUG" or left blank
CC=xlC
CFLAGS_OS=-qflag=i -qlanglvl=ansi -qcpluscmt -DSQLUNIX -DSQLAIX -q64 -O3 -D_LARGE_FILES
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=xlC
LD_STORP=xlC
LD_FLAGS_EXEC=-lm -q64
LD_FLAGS_SHLIB=qmkshrobj
LD_FLAGS_STORP=$(LD_FLAGS_SHLIB) -bE:$@.exp -lc -b64
LD_FLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2
LD_FLAGS_OUT=-o

# Library Configuration
AR=ar
AR_FLAGS=-r -v -X64
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=.a
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
#####

PRP_OPTS = PACKAGE \
           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 = tpcdbg$(OBJEXT)
UTIL_OBJ_GEN = tpcmisc$(OBJEXT)
UTIL_OBJ_DB2 = tpcctx$(OBJEXT)

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

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

dbgen: $(UTIL_OBJ_GEN)

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

#####
# Helper Targets
```

```
#####
connect:
    - db2 connect to $(TPCC_DBNAME)

disconnect:
    - db2 connect reset
    - db2 terminate

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

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

.sqc.c:
    @echo "Prepping $*.sqc"
    db2 prep $*.sqc $(PRP_OPTS)
    db2 grant execute on package TPCCCTX to public

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

# Source
tpcdbg$(OBJEXT): tpcdbg.c
tpcctx$(OBJEXT): tpcctx.c
tpccmisc$(OBJEXT): tpccmisc.c

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

## Src.Common/tpcctx.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.
**
*/

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

#include <string.h>
#include <sqlutil.h>
#include "db2tpcc.h"
#include "tpcdbg.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/tpcdbg.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.
**
*/

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

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

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

        default:
            return;
    }

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

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

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

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

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

```

```

        fprintf(err_fp, "\n");

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

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

        fprintf(err_fp, "\n");

        fclose(err_fp);
    }

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

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

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

        current_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");

        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, "%ts_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, "PID %d ", getpid());
    fprintf(debug_fp, "\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, "%ts_I_NAME[%d] = %s\n",
                j, neword_ptr->item[j].s_I_NAME);
        fprintf(debug_fp, "%ts_I_PRICE[%d] = %d.2f\n",
                j, neword_ptr->item[j].s_I_PRICE);
        fprintf(debug_fp, "%ts_OL_AMOUNT[%d] = %d.2f\n",
                j, neword_ptr->item[j].s_OL_AMOUNT);
        fprintf(debug_fp, "%ts_S_QUANTITY[%d] = %d (%X)\n",
                j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr->item[j].s_S_QUANTITY);
        fprintf(debug_fp, "%ts_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)
{
    InitializeDebug();
    strncpy(debug_in, debugPath, DEBUG_PATH_SIZE);

```

```

j, in_neword->in_item[j].s_OL_SUPPLY_W_ID, in_neword->in_item[j].s_OL_SUPPLY_W_ID);
fprintf(debug_fp, "%ts_OL_QUANTITY[%d] = %d (%X)\n",
        j, in_neword->in_item[j].s_OL_QUANTITY, in_neword->in_item[j].s_OL_QUANTITY);
}
fprintf(debug_fp, "%t\n\n");

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 = %d.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, "%ts_I_NAME[%d] = %s\n",
            j, neword_ptr->item[j].s_I_NAME);
    fprintf(debug_fp, "%ts_I_PRICE[%d] = %d.2f\n",
            j, neword_ptr->item[j].s_I_PRICE);
    fprintf(debug_fp, "%ts_OL_AMOUNT[%d] = %d.2f\n",
            j, neword_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "%ts_S_QUANTITY[%d] = %d (%X)\n",
            j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr->item[j].s_S_QUANTITY);
    fprintf(debug_fp, "%ts_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)
{
    InitializeDebug();
    strncpy(debug_in, 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=====");

    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, "%t\n\n");

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

    fprintf(debug_fp, "%titems {\n");
    items = ordstat_ptr->s_ol_cnt;
    for (j = 0; j < items; j++) {
        if(j != 0)
            fprintf(debug_fp, "\n");
        fprintf(debug_fp, "%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, "lts_OL_AMOUNT[%d] = %2fn",
        j, ordstat_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp, "lts_OL_DELIVERY_D[%d] = %s ln",
        j, ordstat_ptr->item[j].s_OL_DELIVERY_D_time);
}
fprintf(debug_fp, "l")lnln");
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)ln", timeStamp, msg);
    fprintf(debug_fp, " PID %d ", getpid());
    fprintf(debug_fp, "ln=====ln");

    fprintf(debug_fp, "in_payment_struct {ln");
    fprintf(debug_fp, "lts_H_AMOUNT = %2f ln",
        in_payment->s_H_AMOUNT);
    fprintf(debug_fp, "lts_C_ID = %d (%X)ln",
        in_payment->s_C_ID, in_payment->s_C_ID);
    fprintf(debug_fp, "lts_W_ID = %d (%X)ln",
        in_payment->s_W_ID, in_payment->s_W_ID);
    fprintf(debug_fp, "lts_D_ID = %d (%X)ln",
        in_payment->s_D_ID, in_payment->s_D_ID);
    fprintf(debug_fp, "lts_C_D_ID = %d (%X)ln",
        in_payment->s_C_D_ID, in_payment->s_C_D_ID);
    fprintf(debug_fp, "lts_C_W_ID = %d (%X)ln",
        in_payment->s_C_W_ID, in_payment->s_C_W_ID);
    fprintf(debug_fp, "lts_C_LAST = %sln",
        in_payment->s_C_LAST);
    fprintf(debug_fp, "lnln");

    fprintf(debug_fp, "out_payment_struct {ln");
    fprintf(debug_fp, "lts_C_CREDIT_LIM = %2fn",
        payment_ptr->s_C_CREDIT_LIM);
    fprintf(debug_fp, "lts_C_DISCOUNT = %0.4fn",
        payment_ptr->s_C_DISCOUNT);
    fprintf(debug_fp, "lts_C_BALANCE = %2fn",
        payment_ptr->s_C_BALANCE);

```

```

    fprintf(debug_fp, "lts_C_ID = %d (%X)ln",
        payment_ptr->s_C_ID, payment_ptr->s_C_ID);
    fprintf(debug_fp, "lts_W_STREET_1 = %sln",
        payment_ptr->s_W_STREET_1);
    fprintf(debug_fp, "lts_W_STREET_2 = %sln",
        payment_ptr->s_W_STREET_2);
    fprintf(debug_fp, "lts_W_CITY = %sln",
        payment_ptr->s_W_CITY);
    fprintf(debug_fp, "lts_W_STATE = %sln",
        payment_ptr->s_W_STATE);
    fprintf(debug_fp, "lts_W_ZIP = %sln",
        payment_ptr->s_W_ZIP);
    fprintf(debug_fp, "lts_D_STREET_1 = %sln",
        payment_ptr->s_D_STREET_1);
    fprintf(debug_fp, "lts_D_STREET_2 = %sln",
        payment_ptr->s_D_STREET_2);
    fprintf(debug_fp, "lts_D_CITY = %sln",
        payment_ptr->s_D_CITY);
    fprintf(debug_fp, "lts_D_STATE = %sln",
        payment_ptr->s_D_STATE);
    fprintf(debug_fp, "lts_D_ZIP = %sln",
        payment_ptr->s_D_ZIP);
    fprintf(debug_fp, "lts_C_FIRST = %sln",
        payment_ptr->s_C_FIRST);
    fprintf(debug_fp, "lts_C_MIDDLE = %sln",
        payment_ptr->s_C_MIDDLE);
    fprintf(debug_fp, "lts_C_LAST = %sln",
        payment_ptr->s_C_LAST);
    fprintf(debug_fp, "lts_C_STREET_1 = %sln",
        payment_ptr->s_C_STREET_1);
    fprintf(debug_fp, "lts_C_STREET_2 = %sln",
        payment_ptr->s_C_STREET_2);
    fprintf(debug_fp, "lts_C_CITY = %sln",
        payment_ptr->s_C_CITY);
    fprintf(debug_fp, "lts_C_STATE = %sln",
        payment_ptr->s_C_STATE);
    fprintf(debug_fp, "lts_C_ZIP = %sln",
        payment_ptr->s_C_ZIP);
    fprintf(debug_fp, "lts_C_PHONE = %sln",
        payment_ptr->s_C_PHONE);
    fprintf(debug_fp, "lts_C_SINCE = %s ln",
        payment_ptr->s_C_SINCE_time);
    fprintf(debug_fp, "lts_C_CREDIT = %sln",
        payment_ptr->s_C_CREDIT);
    fprintf(debug_fp, "lts_C_DATA = %sln",
        payment_ptr->s_C_DATA);
    fprintf(debug_fp, "lts_transtatus = %d (%X)ln",
        payment_ptr->s_transtatus, payment_ptr->s_transtatus);
    fprintf(debug_fp, "ldeadlocks = %d (%X)ln",
        payment_ptr->deadlocks, payment_ptr->deadlocks);
    fprintf(debug_fp, "lnln");
    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)ln", timeStamp, msg);
    fprintf(debug_fp, " PID %d ", getpid());
    fprintf(debug_fp, "ln=====ln");

    fprintf(debug_fp, "in_stocklev_struct {ln");
    fprintf(debug_fp, "lts_W_ID = %d (%X)ln",
        in_stocklev->s_W_ID, in_stocklev->s_W_ID);
    fprintf(debug_fp, "lts_D_ID = %d (%X)ln",
        in_stocklev->s_D_ID, in_stocklev->s_D_ID);
    fprintf(debug_fp, "lts_threshold = %d (%X)ln",
        in_stocklev->s_threshold, in_stocklev->s_threshold);
    fprintf(debug_fp, "lnln");

    fprintf(debug_fp, "out_stocklev_struct {ln");
    fprintf(debug_fp, "lts_transtatus = %d (%X)ln",
        stocklev->s_transtatus, stocklev->s_transtatus);
    fprintf(debug_fp, "ldeadlocks = %d (%X)ln",
        stocklev->deadlocks, stocklev->deadlocks);
    fprintf(debug_fp, "lts_low_stock = %d (%X)ln",
        stocklev->s_low_stock, stocklev->s_low_stock);
    fprintf(debug_fp, "lnln");
    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 - 2005
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
/*-----*/

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

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

```



```

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

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

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

```

## dbgen/Makefile

```

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

# Makefile - Build gendata tool
#

include $(TPCC_ROOT)/Makefile.config

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

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

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

LDFLAGS = $(LDFLAGS_EXEC) $(LDFLAGS_LIB)

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

OBJS = tpcrnd$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT)
OBJ_EEE = $(TPCC_ROOT)/Src.Common/tpccwh$(OBJEXT)

EXEC = gendata$(BINEXT)

# #####
# End-User Targets
# #####

all: $(EXEC)

clean:
    - $(ERASE) *$(OBJEXT) $(EXEC)

```

```

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

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

# We use $@$(OBJEXT) here so that the UNIX makefiles work with both
# 'traditional' make and GNU make
$(EXEC):
    $(LD_EXEC) $(LDFLAGS) $(OBJS) $@$(OBJEXT) $(LDFLAGS_OUT)$@

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

# Link Dependencies
gendata$(BINEXT): $(OBJS) gendata$(OBJEXT)

# Build Dependencies
# Source
gendata$(OBJEXT): gendata.c

# Headers
gendata.c: $(TPCC_ROOT)/include/tpccrnd.h $(TPCC_ROOT)/include/lval.h

```

## dbgen/gendata.c

```

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

/*
 * gendata.c - Generate data for TPC-C database
 */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sqlutil.h>
/* UNIX named pipe support */
#include <sys/stat.h>
#include <errno.h>
#include <fcntl.h>
#include <time.h>

#include "platform.h"
#include "db2tpcc.h"
#include "tpccrnd.h"
#include "tpccmisc.h"
#include "lval.h"

/* PROTOTYPES. */
void gen_dist_tbl( void );
void gen_cust_tbl( void );
void gen_hist_tbl( void );
void gen_nu_ord_tbl( void );
void gen_ord_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_npice = 0;
int using_rctload = 0;
int quiet_mode = 0;
sqInt32 ware_start=-1, ware_end=-1;

char fmtWare[] = "%s%s%s%s%s%s%04.4f%.2f%d\n";
char fmtDist[] = "%d%04.4f%.2f%s%s%s%s%s%s%04.4f%.2f%d\n";
char fmtItem[] = "%s%04.4f%.2f%d\n";
char fmtStock[] = "%d%04.4f%.2f%d%04.4f%.2f%d\n";
char fmtCust[] = "%d%04.4f%.2f%d%04.4f%.2f%d\n";
char fmtHist[] = "%d%04.4f%.2f%d%04.4f%.2f%d\n";
char fmtOrder[] = "%d%04.4f%.2f%d%04.4f%.2f%d\n";
char fmtOLine[] = "%s%04.4f%.2f%d%04.4f%.2f%d%04.4f%.2f%d\n";
char fmtNewOrd[] = "%d%04.4f%.2f%d\n";
void InitFormatStrings(char delim);
void ScalingReport(void);

int outtype1 = 0;
int outtype2 = 0;
char *outname1 = NULL;
char *outname2 = NULL;

/*-----*/
/* main */
/*-----*/
int main (int argc, char *argv[])
{
    int option = -1;
    char *delim = NULL;

    /* ----- */
    /* Compute Warehouse Ranges */
    /* ----- */
    ware_start = 1;
    ware_end = WAREHOUSES;

    /* ----- */
    /* Process Command Line Arguments */
    /* ----- */

    /* Valid Command Line Options
    *-----
    * Table Option: -t <table> (-t3 for warehouse)
    * Output Column Delimiter: -d <char> (-d'', -d '|', etc)
    * Output to File: -f[n] <file> (-f customer.dat)
    * Output to Pipe: -p[n] <pipe name> (-p tpcpipe.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 tpcpipe.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 inorders/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 (%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_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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    if (!quiet_mode) {
        fprintf(stdout, "\nSTOCK table generated in %.2f seconds.\n\n", elapsed);
        fflush(stdout);
    }
} else {
    fprintf(stderr, "\nSTOCK table FAILED at (S %d W %d) after %.2f seconds.\n\n", stock_num,
    ware_num, elapsed);
    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 %.2f seconds.\n\n", elapsed);
        fflush(stdout);
    }
} else {
    fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after %.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 = 300000.00;

        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 {
    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-rcload) */
        if (using_rcload == 0) {
            for (pos=strlen(cust_data); pos<500; pos++)
                cust_data[pos] = '\0';
            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_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", elapsed);
        fflush(stdout);
    }
} else {
    fprintf(stderr, "nHISTORY table FAILED at (W %d D %d C %d) after %8.2f seconds.\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; }

/* We generate in O/W/D order for non-RCT tables. With the
* data clustered on O_ID, this gives us good bufferpool
* characteristics. We also create a btree index in W/D/O
* order, to satisfy MIN(O_ID) queries.
*
* For RCT tables *with* RCT Jump Cache, we *should* generate
* the data in W/D/O order (to match the table definition.)
* We don't since it would push schema decisions into flat file
* generation (and I don't want to do that.) It's just as easy
* to sort the flat files afterwards.
*/

for (nu_ord_id = nu_ord_hi;
    nu_ord_id <= CUSTOMERS_PER_DISTRICT;
    nu_ord_id++)
{
    if (!quiet_mode) {
        fprintf(stdout, "NEW_ORDER for Customer #:%d:\n", nu_ord_id);
        fflush(stdout);
    }
    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)

```

```

{
    for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
    {
        numBytes = sprintf(Buffer, fmtNewOrd,
            nu_ord_id,
            dist_num,
            ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto neword_done; }

    } /* end for... */
} /* end for... */

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", elapsed);
        fflush(stdout);
    }
} else {
    fprintf(stderr, "nNEW_ORDER table FAILED at (W %d D %d O %d) after %8.2f
seconds.\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 currtmstp[27];
    char nulltmstp[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(currtmstp);

    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, fmtOrd,
        cust_num,
        currtmstp,
        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) ? currtmstp : nulltmstp),
            ((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", 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", ware_num, dist_num, ord_num, oline_ol_num, elapsed);

```

```

flush(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(fmtOrd, '|')) { *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);
    srandom(t);
}

/*
*****
* create_random_a_string
*
* create a random alphanumeric string, of random length between lo and
* hi and place them in designated buffer. Routine returns the actual
* length.
*
* parameters
* -----

```

```

* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random alphanumeric string
*
*
*/

int create_random_a_string( char *out_buffer, int length_lo, int length_hi )
{
    int i, actual_length;

    actual_length = rand_integer( length_lo, length_hi );

    for ( i = 0; i < actual_length; i++ )
    {
        out_buffer[i] = alnum[rand_integer( 0, 61 )];
    }
    out_buffer[actual_length] = '\0';

    return (actual_length);
}

/*
*****
* create_random_n_string
*
* create a random numeric string, of random length between lo and
* hi and place them in designated buffer. Routine returns the actual
* length.
*
* parameters
* -----
* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random numeric string
*
*
*/

int create_random_n_string( char *out_buffer, int length_lo, int length_hi )
{
    int i, actual_length;

    actual_length = rand_integer( length_lo, length_hi );

    for ( i = 0; i < actual_length; i++ )
    {
        out_buffer[i] = (char)rand_integer( 48,57 );
    }
    out_buffer[actual_length] = '\0';

    return (actual_length);
}

/*
*****
* 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 NUrnd_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 );
        strcpy(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 = NUrnd_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
*
*/

#ifdef __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_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_ID 3849
#define A_C_LAST 255
#define A_C_ID 1023
#define A_OL_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_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[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_ol_cnt;
    int16_t pad1[2];
    struct oitems_struct {
        double s_OL_AMOUNT;
        int32_t s_OL_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 20100315.1027 */
```

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 104040
#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 erno
```

```
/* Windows I/O Macros */
```

```
/* UNIX I/O Macros */
```

```
#include <fcntl.h>
```

```
#define IOH_INIT(hnd, type, name)
hnd->fd = -1;
hnd->type = type;
hnd->name = name;
```

```
#define IOH_CREATE(hnd)
if (hnd->type == IOH_PIPE) {
rc = mkfifo(hnd->name, 0666);
} else {
rc = 0;
}
```

```
#define IOH_OPEN(hnd)
```

```
if (hnd->type == IOH_FILE_APPEND) {
hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_APPEND, 0666);
} else {
hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_TRUNC, 0666);
}
if (hnd->fd == -1) {
rc = -1;
} else {
rc = 0;
}
```

```
#define IOH_WRITE(hnd, buff, num, num2)
rc = write(hnd->fd, buff, num);
if (rc >= 0) {
num2 = rc;
rc = 0;
}
```

```
#define IOH_FLUSH(hnd) rc = 0;
#define IOH_CLOSE(hnd) rc = close(hnd->fd);
#define IOH_DELETE(hnd) if (hnd->type == IOH_PIPE) { rc = unlink(hnd->name); }
```

```
typedef unsigned int IOH_NUM;
typedef int IOH_HND;
```

```
/* UNIX Semaphore Macros */
```

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
```

```
union semun {
int val;
struct semid_ds *buf;
unsigned short int *array;
} semUnion;
```

```
struct sembuf semBuf;
```

```
#define SEM_HANDLE int
```

```
#define SEM_INIT(hnd, x, name)
if ( (hnd = semget(IPC_PRIVATE, 1, IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH)) == -1)
API_ERROR(__LINE__, "semget", (rc=GEN_ERRCODE));
semUnion.val = x;
if ( semctl(hnd, 0, SETVAL, semUnion) < 0 )
API_ERROR(__LINE__, "semctl SETVAL", (rc=GEN_ERRCODE));
```

```
#define SEM_WAIT(hnd)
semBuf.sem_num = 0;
semBuf.sem_op = -1;
semBuf.sem_flg = SEM_UNDO;
if ( semop(hnd, &semBuf, 1) < 0 )
API_ERROR(__LINE__, "semop wait", (rc=GEN_ERRCODE));
```

```
#define SEM_FREE(hnd)
semBuf.sem_num = 0;
semBuf.sem_op = 1;
semBuf.sem_flg = SEM_UNDO;
if ( semop(hnd, &semBuf, 1) < 0 )
API_ERROR(__LINE__, "semop free", (rc=GEN_ERRCODE));
```

```
#define SEM_DESTROY(hnd)
if ( semctl(hnd, 0, IPC_RMID, 0)
API_ERROR(__LINE__, "semctl IPC_RMID", (rc=GEN_ERRCODE));
```

```
/* Common I/O Macros and Definitions */
```

```
#define IOH_FILE 1
```

```
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3
```

```
#define IOH_ERRMSG(hnd, msg)
if (rc != 0) {
fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", GEN_ERRCODE, msg,
hnd->fd, hnd->type, hnd->name);
return rc;
}
```

```
struct _ioh {
IOH_HND fd;
int type;
char *name;
};
```

```
typedef struct _ioh ioHandle;
```

```
/* Generic I/O Routine Prototypes */
```

```
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes);
int GenericClose(ioHandle *hnd);
```

```
/* Generic I/O Routines */
```

```
int GenericOpen(ioHandle *hnd, int type, char *name)
{
int rc = 0;
```

```
IOH_INIT(hnd, type, name)
```

```
IOH_CREATE(hnd)
IOH_ERRMSG(hnd, "creating")
```

```
IOH_OPEN(hnd)
IOH_ERRMSG(hnd, "opening")
```

```
return rc;
```

```
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes)
```

```
{
int rc = 0;
int numBytesWritten = -1;
```

```
IOH_WRITE(hnd, Buffer, numBytes, numBytesWritten)
```

```
IOH_ERRMSG(hnd, "writing")
```

```
if (numBytes != numBytesWritten) {
fprintf(stderr, "Truncated data writing to fd %d (%d, %s)\n", hnd->fd, hnd->type, hnd->name);
rc = -1;
}
```

```
return rc;
```

```
int GenericClose(ioHandle *hnd)
```

```
{
int rc = 0;
```

```
IOH_FLUSH(hnd)
IOH_ERRMSG(hnd, "flushing")
```

```
IOH_CLOSE(hnd)
IOH_ERRMSG(hnd, "closing")
```

```
IOH_DELETE(hnd)
IOH_ERRMSG(hnd, "deleting")
```

```
return rc;
```

```
}
```

```
#endif // __PLATFORM_H
```

## include/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 - 2005
** 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"
```

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

```
/*
FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int i=0x12345678; SWAP_BYTE(i); i => 0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by XOR op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Add + 4 - 0 - 1 = Addr+3];
      *a ^= *b; // sets *a to 0x6A
      *b ^= *a; // sets *b to 0x12
      *a ^= *b; // sets *a to 0x78
*/
```

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

```
void SwapEndian(void *Addr, int nb)
```

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

```
    *a ^= *b;
    *b ^= *a;
    *a ^= *b;
```

```
  }
}
```

```
#endif //SWAP_ENDIAN
```

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

```
/*
** 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.
*/

#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
```

## 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=CK080131
```

```
# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}
```

```
# The OS being used (i.e. "UNIX", "LINUX", "WINDOWS")
export PLATFORM=UNIX
export SERVER_PLATFORM=UNIX
```

```
# The type of make command and slash used by the OS.
# (i.e. UNIX - "r", WINDOWS - "\").
# These are referenced all over the kit.
export SLASH="r";
export MAKE=make
```

```
# Specifies whether or not to use dari stored proc's for the TPC-C driver. Set to either DARVERSION or
NONDAR;
#export TPCC_SPTYPE=NOSP
#export TPCC_SPTYPE=SPGENERAL2
export TPCC_SPTYPE=SPGENERAL
#export TPCC_SPTYPE=DARI2SQLDA
```

```
export DB2VERSION=v9
```

```
# The schema name is typically the SQL authorization ID (or username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}
export SERVER_TPCC_SCHEMA=${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
```

# Appendix - D: Pricing Information

CDW CDW-G CDW Canada 800.750.4239 Shopping Cart 1 Items Support Log On

---

CDW
Shop CDW My Account Print This Page

Search for...  All Products Find It Browse All Categories

Products Services Solutions Center What CDW Offers

---

Shopping Cart

[Saved Carts](#)
[Save This Cart](#)
[E-mail This Cart](#)

Quantity	Product	CDW	Availability	Price	Ext. Price
<input type="text" value="1"/>	<a href="#">3Com Baseline Switch 2824 - switch - 24 ports</a>	<b>1931828</b>	In Stock	\$296.99	\$296.99
Click  to remove an item from your cart				Sub-Total	\$296.99

Update Cart
Clear Cart

Use Standard Checkout
Use Express Checkout

Continue Shopping

**Shipping Calc:**

Enter a postal code to quickly estimate shipping cost.

**QuickCart:**

Enter a **CDW part number** to quickly add it to your cart.

Product ID  
 CDW Part: XXXXXXXX  
 Mfg. Part: XXXXXXXXX  
 UHSPSC: XXXXXXXXX

Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052-6399

Tel 425 882 8080  
Fax 425 936 7329  
<http://www.microsoft.com/>

**Microsoft**

April 2, 2010

IBM Corporation  
Lotus Douglas  
11501 Burnet Road  
Austin, TX 78758

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
LWA-00984	<b>Windows Web Server 2008 R2</b> <i>Full License</i> <i>No Discounts Applied</i>	\$469	1	\$469
127-00166	<b>Microsoft Visual Studio 2008 Professional</b> <i>Full License</i> <i>No Discounts Applied</i>	\$799	1	\$799
N/A	<b>Microsoft Problem Resolution Services</b> <i>Professional Support</i> <i>(1 Incident)</i>	\$259	1	\$259

All products are currently orderable through Microsoft's normal distribution channels. A list of Microsoft's resellers can be found at <http://www.microsoft.com/products/info/render.aspx?view=22&type=how>.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$259 per call.

This quote is valid for the next 90 days.

Reference ID: PCLoDo10040200061426.



11400 BURNET RD  
AUSTIN TX 78758

International Business Machines Corporation

April 7, 2010

Dear Lotus,

Here is the requested quote for the System IBM Power 780 Server TPC-C benchmark using DB2 9.5 and IBM System Storage DS3400.

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
<b>Server Hardware</b>						
Server 1:9179-MHB Base MTM	9179-MHB	1	10,195	1	10,195	8,315
IBM MMB DRWR, IBM BEZEL +CHASSIS/IBM LABELS+	5597	1	12,000	1	12,000	
AC POWER SUPPLY, 200-240V, 1725 WATT	5632	1	1,502	2	3,004	
F8P1 FLEXIBLE SERVICE PROCESSOR CARD	5664	1	4,000	1	4,000	
QUAD (2X RJ45 1GB / 2X SFP+ 10GB) HEA	1803	1	699	1	699	
GX++ DUAL-PORT IB ADPTR	1808	1	1,499	2	2,998	
1.5 Meter 12X DDR Cable	1862	1	524	8	4,192	
SAS Cable (X) Adapter to SAS Enclosure	3661	1	197	1	197	
PWR CBL, DRWR TO IBM PDU, 9' 200-240V/10A	6671	1	19	8	152	
IBM/OEM Rack MTG RAIL kit, ADJUSTABLE DEPTH	7164	1	222	2	444	
OPERATOR PANEL + SHIP GROUP, P7 MR	1853	1	1,000	1	1,000	
3.8 / 4.1GHZ, 0'16w Core POWER7, 16 DDR3 Memory Slots	4982	1	57,429	1	57,429	6,072
1W PROCESSORACTIVATION FOR FC 4982	5469	1	8,375	8	67,000	23,040
0'128GB/4X32GB) SDRAM DDR3 DIMMS, 1066MHZ	5602	1	15,440	4	61,760	
1GB DDR3 MEMORY ACTIVATION	8212	1	245	512	125,440	
146GB 15K RPM SFF SAS DISK	1886	1	1,045	2	2,090	
System AC Power Supply, 1725 w	5632	1	1,652	2	3,304	
DISK/MEDIA BACK PLANE, 6X SFF DISK BAYS, 1X SA	5652	1	4,000	1	4,000	
SATA Slimline DVD-RAM Drive	5762	1	392	1	392	
12X I/O DRAWER PCIE, SFF Disk	5902	1	14,277	3	42,831	23,400
4 Gigabit Fibre Channel dual-port PCI-X Adapter (Jack)	5759	1	3,308	1	3,308	
IBM 2-port 10/100/1000 Base-TX PCI Express Adapter	5767	1	692	1	692	
Power Control Cable (SPCN) - 3 Meter	6006	1	52	4	208	
Dual Port 12X Channel Attach - Short Hun	6446	1	755	1	755	
I/O Drawer Mounting Enclosure	7314	1	687	1	687	
Front Door OEM (Black)	6246	1	970	1	970	
Side Panel (Black)	6098	1	150	2	300	
PDU to 14', 200-240V/24A, UTG0247, PT#12	6654	1	240	2	480	
Power Dist Unit-Side Mount, Universal UTG0247	7188	1	1,000	2	2,000	
HMC 1:7042-C07 Desktop Hardw.Mgmt.Console	7042-C07	1	1,830	1	1,830	1,344
IBM T117 FLAT PANEL MONITOR	3645	1	875	1	875	
Power Cord (6-foot), To Wall Plug Type #4	6470	1	18	2	36	
Ethernet Cable, 6M, HMC to System Unit	7801	1	15	1	15	
Keyboard - English, #103P	5951	1	107	1	107	
USB Mouse	8845	1	39	1	39	
3.5 TB SSD Package	FC4367	1	98,114	3	294,342	50,400
IBM 36U Enterprise Rack	7014-T00	1	2,920	1	2,920	768
DPI Single-phase 30A/208V C19 Enterprise PDU (US)	6062	1	689	2	1,378	
				<b>Subtotal</b>	<b>714,029</b>	<b>113,339</b>
<b>Storage</b>						
DS3400	1726-42E	1	9,292	1	9,292	1,300
EXP3000	1727-01X-2676	1	3,199	3	9,597	2,280
1 TB 7.2K rpm SATA 1000GB	1727-01X-5561	1	1,199	26	31,174	
300GB HS 15K SAS HDD	5532	1	699	8	5,592	
SAS Cables 0.6 Meters	3688	1	119	4	476	
Fiber Cable 1m	5601	1	79	2	158	
				<b>Subtotal</b>	<b>56,289</b>	<b>3,580</b>
<b>Server Software</b>						
ADX V6 (media only)	5692-A6P	1	50	1	50	
ADX 6 for POWER V6.1	5765-G62	1	2,600	8	20,800	
ADX per processor SWMA Large Power 7 (3Y)	5773-SM3-1260	1	1,755	8		14,040
ADX per processor SWMA Large Power 7 24x7 Upgrade (3Y)	5773-SM3-1261	1	718	8		5,744
HMC Software SUB (3Y)	5773-0570	1	461	1	461	
HMC Software Support (3Y)	5773-0569	1	675	1	675	
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	1,140	1	1,140	
C for AIX user annual SW maint renewal	E1A1FLL	1	228	2	456	
DB2 9.5 Enterprise Proc Lic/1 year Maintenance		1	405	960	388,800	
DB2 9.5 Enterprise Edition Proc Maint Renew		1	81	1,920		155,520
				<b>Subtotal</b>	<b>410,790</b>	<b>176,896</b>

<b>Client Hardware and Software</b>						
IBM System x3650 M2 (Quad-core Xeon 2.4GHz)	7946AC1	1	3,316	8	26,528	
1 GB memory	3963	1	85	24	2,040	
146GB 10K RPM SAS SFF	5537	1	269	8	2,152	
Optical 3-Button Mouse - USB	8913	1	19	1	19	
Preferred Pro Full Size PS/2 Keyboard	40K9584	1	29	1	29	
ServicePac for 3-Year 24x7x4 Support	6756298	1	450	8		3,600
IBM T115 15" TFT Monitor	494215U	1	209	1	209	
			<b>Subtotal</b>		<b>30,977</b>	<b>3,600</b>
<b>Total</b>					<b>1,212,085</b>	<b>297,415</b>
<b>Total IBM Discounts*</b>						<b>-689,848</b>
<b>Three-Year Cost of Ownership</b>						<b>819,652</b>

For more information on:

All products: [ibm.com/products](http://ibm.com/products)

For Storage products: [ibm.com/systems/storage/](http://ibm.com/systems/storage/)

For Power System products: [ibm.com/systems/power/](http://ibm.com/systems/power/)

For DB2: <http://www-306.ibm.com/software/data/db2/9/>

For additional information, please contact me directly:

Dan Hebrank

IBM

Director, Power Systems, Americas

Office: 314-252-4160