
IBM eServer p5 595

Model 9119-595

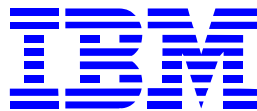
Using

AIX 5L Version 5.3

and

DB2 Universal Database 8.2

TPC BenchmarkTM C
Full Disclosure Report



Third Edition

April 10, 2005

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 eServer pSeries

IBM eServer xSeries

AIX

IBM

DB2, DB2 UDB, DB2 Universal Database

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 2000 server and COM+ are registered trademarks of Microsoft Corporation

First Edition: November 18, 2004

Second Edition: November 18, 2005

Third Edition: April 10, 2006

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, 2004. 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.

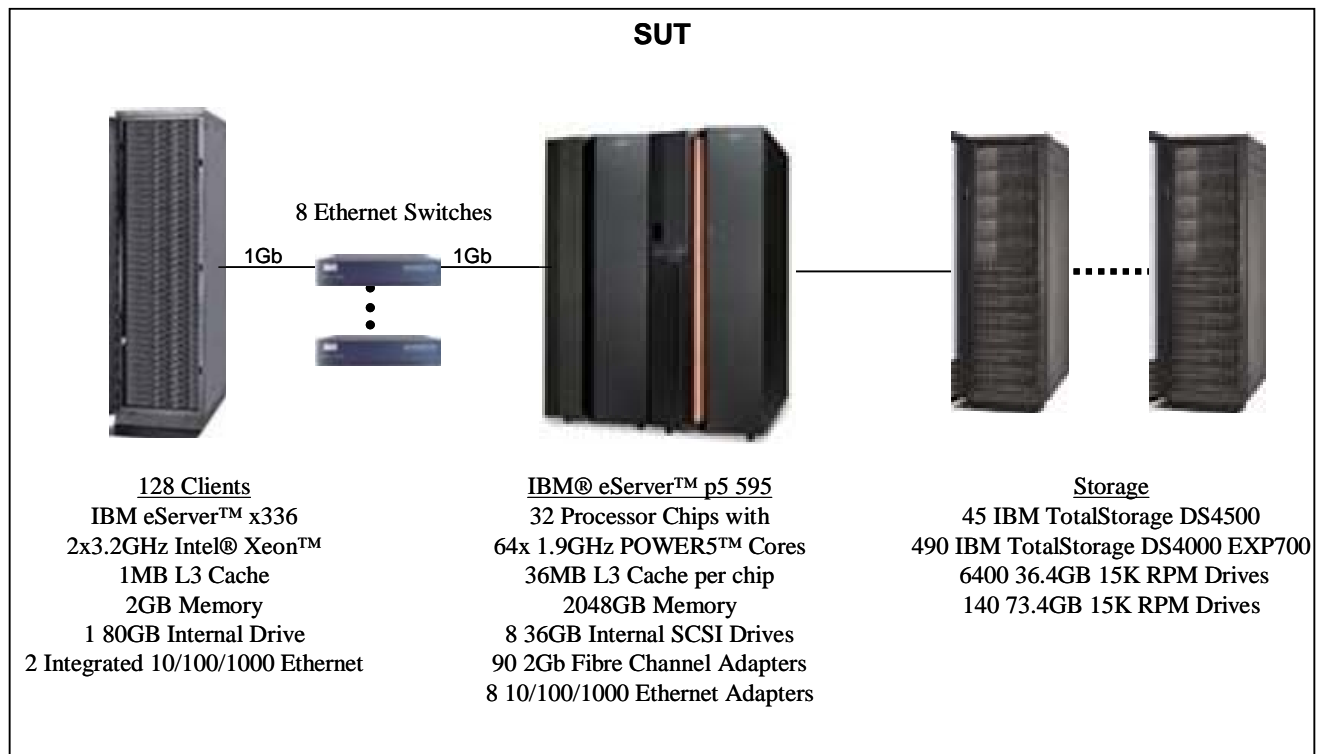


**IBM eServer p5 595
Model 9119-595**

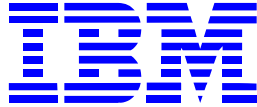
TPC-C Rev. 5.3

Report Date: November 18, 2004
Updated: April 10, 2006

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
\$16,273,320 USD	3,210,540.63	\$5.07 USD	May 14, 2005	
Processor Chip/Core/Thread	Database Manager	Operating System	Other Software	No. Users
32/64/128	DB2 UDB 8.2	AIX 5L V5.3	Microsoft Visual C++ Microsoft COM+	2,560,000



System Components	Each of the 128 Clients		Server	
	Quantity	Description	Quantity	Description
Processor Chips/Cores/Threads	2/2/2	3.2GHz 1MB L3 Xeon Processor	32/64/128	1.9GHz POWER5™
Memory	4	512 MB	4	512 GB
Disk Controllers	1	SATA	2 90 45	Integrated dual Ultra3SCSI 2Gb FC Adapters IBM DS4500 Controllers
Disk Drives	1	80 GB	6400 140 8	36.4GB 15K RPM FC 73.4GB 15K RPM FC 36.4GB 15K RPM SCSI
Total Storage		10240 GB		243,236GB
Terminals	4	System Console	1	System Console



IBM eServer p5 595 Model 9119-595

TPC-C Rev. 5.3

Report Date: November 18, 2004

Update: November 18, 2005

Description	Part No.	Src	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
Server 1:- p5 595	9119-595	1	92,000	1	92,000	63,000
RIO-2 (Remote I/O-2) Cable, 3.5M	3147	1	660	16	10,560	
RIO-2(Remote I/O-2)Cbl, 8.0M	3170	1	800	16	12,800	
36.4 GB 15,000 RPM Ultra320 SCSI Disk Drive Assembly	3277	1	2,125	8	17,000	
2 Gigabit Fibre Channel PCI-X Adapter	5716	1	2,720	90	244,800	
IBM 10/100/1000 Base-TX Ethernet PCI-X Adapter	5701	1	1,280	8	10,240	
PCI-X Dual Channel Ultra320 SCSI Blind Swap	5710	1	790	1	790	
Expansion Rack, Powered	5792	1	60,000	1	60,000	3,960
I/O Drawer, 20 Slots, 8 Disk Bays	5794	1	30,000	8	240,000	32,640
I/O Drw.Cbl.Grp, Prim.Rck/9U	6121	1	400	1	400	
I/O Drw.Cbl.Grp, Prim.Rck/5U	6122	1	400	1	400	
I/O Drw.Cbl.Grp, Prim.Rck/1U	6123	1	400	1	400	
I/O Drw.Cbl.Grp, Prim.Rck/13U	6124	1	400	1	400	
Bulk Power Regulator	6186	1	4,000	8	32,000	
Slim Line Doors, System and #5792 Racks	6251	1	6,000	2	12,000	
Power Cord - United States/Canada	6470	1	17	1	17	
Ethernet Cable, 6M, HMC to System Unit	7801	1	14	2	28	
Power Controller Assembly	7803	1	4,000	4	16,000	
Cooling Group, 2-4 Processor Books	7807	1	4,000	1	4,000	
DC Power Converter, Processor Book	7809	1	6,000	12	72,000	
Processor Clock Card, Programmable	7810	1	575	2	1,150	
System Service Processor	7811	1	3,500	2	7,000	
Multiplexer Card	7812	1	2,200	4	8,800	
16-Way POWER5 Turbo CUoD Processor, 0-Way Active	7813	1	127,000	4	508,000	68,640
Activation, #7813 CUoD Processor Book, One Processor	7815	1	32,000	64	2,048,000	506,880
Remote I/O-2 (RIO-2) Loop Adapter, Two Port	7818	1	3,400	16	54,400	
Pwr.Cbl.Grp, CEC Primary Fans	7821	1	650	1	650	
Pwr.Cbl.Grp, 1st CEC Book	7822	1	650	1	650	
Pwr.Cbl.Grp, 2nd CEC Book	7823	1	650	1	650	
Pwr.Cbl.Grp, 3rd CEC Book	7824	1	650	1	650	
Pwr.Cbl.Grp, 4th CEC Book	7825	1	650	1	650	
Pwr.Cbl.Grp, 7807 Cooling Grp.	7826	1	650	1	650	
Power Distribution Assembly	7837	1	2,500	6	15,000	
I/O Drw.Cbl.Grp, 5792 Rck/1U	7853	1	650	1	650	
I/O Drw.Cbl.Grp, 5792 Rck/5U	7854	1	650	1	650	
I/O Drw.Cbl.Grp, 5792 Rck/9U	7855	1	650	1	650	
I/O Drw.Cbl.Grp, 5792 Rck/19U	7857	1	400	1	400	
512GB DDR1 Memory (16 X 32GB Cards)	8198	1	2,228,224	4	8,912,896	
Line Cord, 6AWG, 14ft, IEC309 100A Plug	8686	1	2,500	4	10,000	
Storage Device Enclosure	7212-102	1	1,295	1	1,295	264
DVD-ROM Drive	1106	1	500	1	500	72
Drive Panel (Blank)	1204	1	25	1	25	
Rack Mounting Hardware Kit	1205	1	150	1	150	
Rack Power Cord, Single	5060	1	29	1	29	
HMC 1:7310-C03 Desktop Hardw.Mgmt.Console	7310-C03	1	2,001	1	2,001	816
Subtotal					12,401,331	676,272
Storage						
DS4500 Disk System	1742-90U	1	49,900	45	2,245,500	
DS4000 EXP700 Storage Expansion	1740-1RU	1	6,000	490	2,940,000	
2Gb FC, 36.4GB/15K Drive	5212	1	1,115	6,400	7,136,000	
2Gb FC, 73.4GB/15K Disk Module	5213	1	2,099	140	293,860	
Short Wave SFP	2210	1	499	1,640	818,360	
Fiber Cable 25m	5625	1	189	90	17,010	
Fiber Cable 1m	5601	1	79	900	71,100	



IBM eServer p5 595 Model 9119-595

TPC-C Rev. 5.3

Report Date: November 18, 2004
Updated November 18, 2005

Fiber Cable 5m	5605	1	129	80	10,320	
2Gb Mini HUB	3507	1	899	90	80,910	
Storage Manager V8.4 upgrade	1742-7109	1	2,999	5	14,995	
3 Year Warranty Service Upgrade 1740-1RU 24x7x4	41L2768	1	760	490		372,400
3 Year Warranty Service Upgrade 1742-90U 24x7x4	96P2062	1	1,087	45		48,915
			Subtotal		13,628,055	421,315
Server Software						
AIX 5.3 (media only)	5692-A5L	1	50	1	50	
AIX Software per Processor	5765-G03	1	2,495	64	159,680	
AIX Software Maintenance (3Y)	5773-SM3	1	2,836	64		181,504
PLM Software Maintenance (3Y)	5773-PL3	1	35	64		2,240
VIO Software Maintenance (3Y)	5773-PL3	1	155	64		9,920
AIX Software Maintenance 24x7 Upgrade (3Y)	0468	1	496	64		31,744
HMC Software SUB (3Y)	5773-0570	1	236	1		236
HMC Software Support (3Y)	5773-0569	1	675	1		675
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	515	1	515	
C for AIX user annual SW maint renewal	E1A1FLL	1	103	2		206
DB2 Enterprise Server Edition Proc Lic/1 yr Maint.	D518GLL	1	21,425	64	1,371,200	
DB2 Enterprise Server Ed Proc Maint Renew	E00BILL	1	1,020	128		130,560
			Subtotal		1,531,445	357,085
Client Hardware and Software						
	3,992					
x336 w 3.2GHz/1024KB Xeon	883722U	1	2,529	128	323,712	57,600
xSeries 3.2GHz 800MHz 1MB L2Cache Intel Xeon	13N0663	1	899	128	115,072	
1GB PC2-3200 (2x512MB) ECC DDR2 SDRAM RDIMM	73P3522	1	399	128	51,072	
80GB 7200 RPM SATA HDD with EZ swap Tray	73P8002	1	165	128	21,120	
Microsoft Windows 2000 Server	C11-00821	3	738	128	94,464	
Microsoft Problem Resolution Services		3	245	1		245
NetBay42 Standard Rack	9306-421	1	1,439	46	66,194	8,648
xSeries Cable Chain Technology Cable Kit	06P4792	1	54	4	216	
IBM Sleek 2-Button Mouse - (PS/2)	28L3673	1	15	4	60	
Preferred Pro Full Size PS/2 Keyboard	31P7415	1	29	4	116	
E54 15" monitor	W9SP47N	1	119	4	476	
			Subtotal		672,502	66,493
Third Party Hardware/Software						
Visual C++ Standard Edition	254-00170	3	109	1	109	
Cisco Catalyst 2970 24 10/100/1000 BASE-T ports	511987	2	2,809	10	28,094	
			Subtotal		28,203	
			Total		28,261,536	1,521,165
					IBM Total System Discounts*	-13,509,381
					Three-Year Cost of Ownership	16,273,320
					tpmC	3,210,540.6
					\$/tpmC	5.07

*Discounts are based on US list prices for similar quantities & configurations including pre-payment for maintenance

Audited by: Francois Raab, Info Sizing (www.infosizing.com)

Pricing Sources:

1 IBM: Bill Casey, eServer pSeries Offering Manager, wr Casey@us.ibm.com, 512-838-1422

1 IBM DB2: Paul Rivot, Director Database Servers Business Intelligence Software, email privot@us.ibm.com, phone 914-766-1325

2. CDW.com, 3. Microsoft Corporation

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. Thank you

Numerical Quantities Summary for the IBM eServer p5 595 Model 9119-595

MQTH, computed Maximum Qualified Throughput: 3,210,540.63 tpmC

<u>Response Times (in seconds)</u>	<u>90th %</u>	<u>Average</u>	<u>Maximum</u>
New Order	0.53	0.34	6.84
Payment	0.53	0.34	7.17
Order-Status	0.53	0.34	6.90
Delivery (interactive)	0.27	0.16	6.50
Delivery (deferred)	0.20	0.15	0.
Stock-Level	0.53	0.35	6.16
Menu	0.25	0.16	6.51

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	44.90%
Payment	43.03%
Order-Status	4.01%
Delivery	4.02%
Stock-Level	4.02%

<u>Keying/Think Times (in seconds)</u>	<u>Min.</u>	<u>Average</u>	<u>Max.</u>
New Order	18.00/0.01	18.01/12.02	18.06/120.23
Payment	3.00/0.01	3.01/12.02	3.06/120.22
Order-Status	2.00/0.01	2.01/10.01	2.05/100.11
Delivery	2.00/0.01	2.01/5.02	2.05/50.23
Stock-Level	2.00/0.01	2.01/5.02	2.05/50.21

Test Duration

Ramp-up Time	44 minutes
Measurement interval	2 hours 30 minutes
Transactions during measurement interval (all types)	1,072,383,088
Ramp-down time	20 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.....	15
2.12. Queuing Mechanism of Delivery.....	15
3 Clause 3: Transaction and System Properties	15
3.1. Atomicity Requirements.....	15
3.2. Consistency Requirements	15
3.3. Isolation Requirements.....	15
3.4. Durability Requirements.....	15
4 Clause 4: Scaling and Data Base Population Related Items	15
4.1. Cardinality of Tables	15
4.2. Distribution of Tables and Logs	15
4.3. Data Base Model Implemented	15
4.4. Partitions/Replications Mapping	15
4.5. 60-Day Space Calculations.....	15
5 Clause 5: Performance Metrics and Response Time Related Items.....	15
5.1. Response Times.....	15
5.2. Keying and Think Times	15
5.3. Response Time Frequency Distribution	15
5.4. Performance Curve for Response Time versus Throughput.....	15
5.5. Think Time Frequency Distribution	15
5.6. Throughput versus Elapsed Time	15
5.7. Steady State Determination	15
5.8. Work Performed During Steady State	15
5.9. Measurement Interval.....	15
6 Clause 6: SUT, Driver, and Communication Definition Related Items.....	15
6.1. RTE Availability	15
6.2. Functionality and Performance of Emulated Components	15
6.3. Network Bandwidth.....	15
6.4. Operator Intervention	15
7 Clause 7: Pricing Related Items.....	15
7.1. Hardware and Programs Used	15
7.2. Three Year Cost of System Configuration	15
7.3. Availability Dates.....	15
7.4. Statement of tpmC and Price/Performance.....	15

8	Clause 9: Audit Related Items	15
	Appendix - A: Client Server Code	15
	A.1 Client/Terminal Handler Code	15
	A.2 Client Transaction Code	15
	Appendix - B: Tunable Parameters	15
	B.1 Database Parameters	15
	B.2 Transaction Monitor Parameters	15
	B.3 AIX Parameters	15
	Appendix - C: Database Setup Code	15
	C.1 Database Creation Scripts	15
	C.2 Data Generation Code	15
	Appendix - D: RTE Scripts	15
	D.1 RTE Parameters	15
	D.2 RTE Scripts	15
	Appendix - E: Third Party Pricing Information	15

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.3 dated April, 2004, for measurements on the IBM eServer p5 595 Model 9119-595. The software used on the IBM eServer p5 595 Model 9119-595 includes AIX 5L Version 5.3 operating system, DB2 UDB 8.2 database manager. Microsoft COM+ is used as transaction manager.

IBM eServer p5 595 Model 9119-595

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM eServer p5 595 Model 9119-595	DB2 UDB 8.2	AIX 5L Version 5.3

Total System Cost	TPC-C Throughput	Price/Performance
<ul style="list-style-type: none">• Hardware• Software• 3 Years Maintenance	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC
\$16,273,320 USD	3,210,540.63	\$5.07 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 5.3 in April 2004.

This is the full disclosure report for benchmark testing of the IBM eServer p5 595 Model 9119-595 and DB2 UDB 8.2 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 eServer pSeries application code for the five TPC Benchmark™ C transactions. Appendix D contains the terminal functions and layouts.

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.*

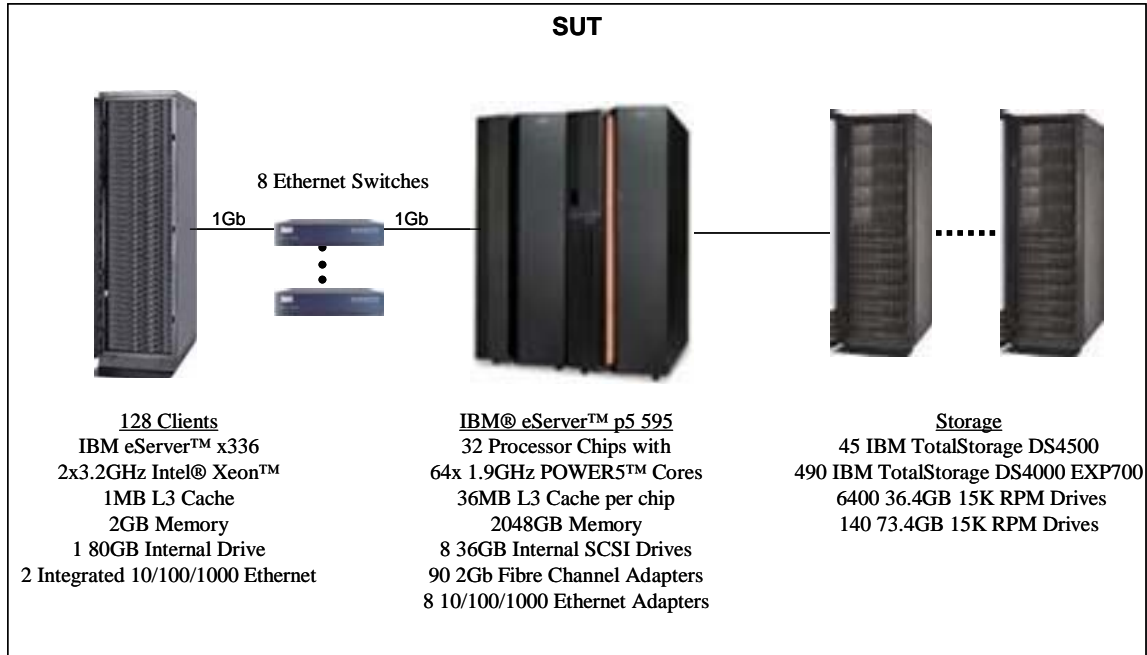
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

0.4. Configuration Diagrams

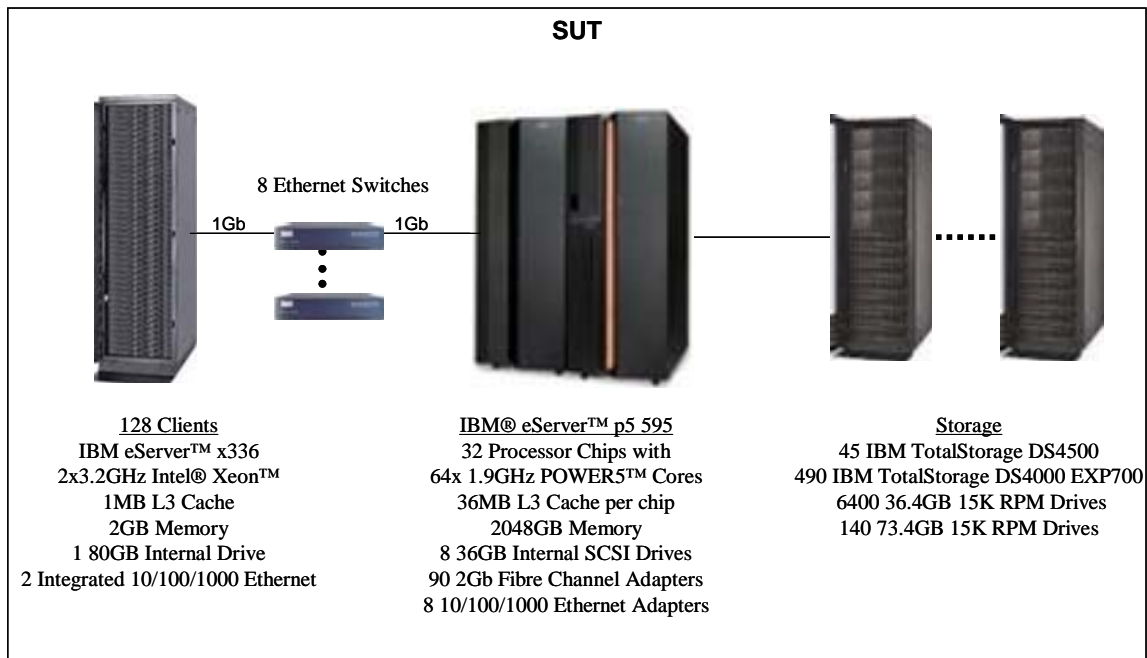
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *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 eServer p5 595 Model 9119-595 Benchmark Configuration



IBM eServer p5 595 Model 9119-595 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 UDB 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 UDB 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 (STOCK, CUSTOMER, ORDERS, ORDERLINE) table partition contains data associated with a range of 1600 warehouses.

Each (WAREHOUSE, DISTRICT, NEWORDER) table partition contains data associated with a range of 32,000 warehouses.

Each HISTORY table partition contains data associated with a range of 16,000 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 eServer x335 systems, are commercially available and support all of the requirements in Clause 2.2.2.4.

2.4. Presentation Managers

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

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

2.5. Home and Remote Order-lines

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

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

2.6. New-Order Rollback Transactions

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

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

2.7. Number of Items per Order

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

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

2.8. Home and Remote Payment Transactions

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

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

2.9. Non-Primary Key Transactions

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

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

2.10. Skipped Delivery Transactions

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

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

2.11. Mix of Transaction Types

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

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

New Order	IBM eServer p5 595 Model 9119-595
Percentage of Home order lines	99.01%
Percentage of Remote order lines	0.99%
Percentage of Rolled Back Transactions	0.99%
Average Number of Items per order	10.00
Payment	
Percentage of Home transactions	85.00%
Percentage of Remote transactions	15.00%
Non-Primary Key Access	
Percentage of Payment using C_LAST	59.99%
Percentage of Order-Status using C_LAST	60.00%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.90%
Payment	43.03%
Order-Status	4.01%
Delivery	4.02%
Stock-Level	4.02%

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 full load test was started and allowed to run for over 10 minutes.
3. One of the disks containing the transaction log was removed. Since the disk was RAID-5, DB2 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 Fibre Channel 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 without write-back cache.
7. The system was subsequently shut down.

8. The disk from step 3 was replaced.
9. The system was powered back on and DB2 was allowed to recover.
10. Step 1 was performed returning the value for SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.

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 25% of the full load.
4. A disk containing the TPCC table was removed causing DB2 to report numerous errors when attempting to access that device
5. The disk was powered back on and the full database was restored from the backup copy in step 1.
6. DB2 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 SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.
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 allowed to run for over 10 minutes.
3. The system was powered off, which removed power from all system components, including memory.
4. The system was powered back on and DB2 recovered.
5. Step 1 was performed returning SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure
6. Consistency condition 3 was verified.

4 Clause 4: Scaling and Data Base Population Related Items

4.1. Cardinality of Tables

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

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

All tables are based on 256,000 warehouses, the number of active warehouses during the benchmark.

Table Name	Number of Rows
Warehouse	256,000
District	2,560,000
Customer	7,680,000,000
History	7,680,000,000
Order	7,680,000,000
New Order	2,304,000,000
Order Line	76,800,279,680
Stock	25,600,000,000
Item	100,000

Table 4-1: Initial Cardinality of Tables

4.2. Distribution of Tables and Logs

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

There is one Logical Volume (LV) for the logs.

There are 5 pairs of storage adapters for the log LV. Each adapter has 2 logical disks (hdisks) and the log LV is striped across the 10 hdisks.

Each hdisk is configured as a RAID5 disk array, with 14 physical disks.

There is a total of 140 physical disks for the logs, and each physical disk has a capacity of 36.4 GB

There are 1920 Logical Volumes (LV) for the tables.

There are 40 pairs of storage adapters for the tables. Each adapter has 4 hdisks and each LV is spread across 2 hdisks (1 hdisk on each adapter of a pair).

There is a total of 48 LVs for each adapter pair.

Each hdisk is configured as a RAID0 disk array, with 20 physical disks.

There is a total of 6400 data disks and each physical disk has a capacity of 36.4 GB drives.

Each LV correspond to one DB2 container.

Each partition within a partitioned table is made of either:

- one DB2 container (STOCK, CUSTOMER, ORDERS, ORDERLINE),
- 20 DB2 containers (WAREHOUSE, DISTRICT, NEWORDER), or
- 10 DB2 containers (HISTORY)

so that the corresponding view spans all 80 adapters (ie all 320 hdisks).

The ITEM table, which is not partitioned, is made of 160 DB2 containers and also spans all 80 adapters.

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 UDB 8.2. DB2 UDB 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 Table 4-2.

DATA DISTRIBUTION		
HDISK	ADAPTER	LOGICAL VOLUMES
hdisk307 hdisk308	fcs72, fcs83	db2loglv
hdisk138 hdisk139	fcs25, fcs43	
hdisk56, hdisk57	fcs3, fcs15	
hdisk224 hdisk225	fcs50, fcs61	
hdisk142 hdisk143	fcs27, fcs45	
hdisk48	fcs4	D1F01V1ITEM, D1F01V1WAE, D1F01V1DIST, D1F01V1CSTI, D1F01V1NORA, D1F01V1ORL, D1F01V1STK, D1F01V1CST, D1F01V1ORDI, D1F01V1ORD, D1F01V1HIST, D1F01V1NORB
hdisk49	fcs1	
hdisk50	fcs4	D1F01V2ITEM, D1F01V2WARE, D1F01V2DIST, D1F01V2CSTI, D1F01V2NORA, D1F01V2ORL, D1F01V2STK, D1F01V2CST, D1F01V2ORDI, D1F01V2ORD, D1F01V2HIST, D1F01V2NORB
hdisk51	fcs1	
hdisk52	fcs4	D1F01V3ITEM, D1F01V3WARE, D1F01V3DIST, D1F01V3CSTI, D1F01V3NORA, D1F01V3ORL, D1F01V3STK, D1F01V3CST, D1F01V3ORDI, D1F01V3ORD, D1F01V3HIST, D1F01V3NORB
hdisk53	fcs1	
hdisk54	fcs4	D1F01V4ITEM, D1F01V4WARE, D1F01V4DIST, D1F01V4CSTI, D1F01V4NORA, D1F01V4ORL, D1F01V4STK, D1F01V4CST, D1F01V4ORDI, D1F01V4ORD, D1F01V4HIST, D1F01V4NORB
hdisk55	fcs1	
hdisk40	fcs2	D1F02V1ITEM, D1F02V1WARE, D1F02V1DIST, D1F02V1CSTI, D1F02V1NORA, D1F02V1ORL, D1F02V1STK, D1F02V1CST, D1F02V1ORDI, D1F02V1ORD, D1F02V1HIST, D1F02V1NORB
hdisk41	fcs0	
hdisk42	fcs2	D1F02V2ITEM, D1F02V2WARE, D1F02V2DIST, D1F02V2CSTI, D1F02V2NORA, D1F02V2ORL, D1F02V2STK, D1F02V2CST, D1F02V2ORDI, D1F02V2ORD, D1F02V2HIST, D1F02V2NORB
hdisk43	fcs0	
hdisk44	fcs2	D1F02V3ITEM, D1F02V3WARE, D1F02V3DIST, D1F02V3CSTI, D1F02V3NORA, D1F02V3ORL, D1F02V3STK, D1F02V3CST, D1F02V3ORDI, D1F02V3ORD, D1F02V3HIST, D1F02V3NORB
hdisk45	fcs0	
hdisk46	fcs2	D1F02V4ITEM, D1F02V4WARE, D1F02V4DIST, D1F02V4CSTI, D1F02V4NORA, D1F02V4ORL, D1F02V4STK, D1F02V4CST, D1F02V4ORDI, D1F02V4ORD, D1F02V4HIST, D1F02V4NORB
hdisk47	fcs0	
hdisk66	fcs6	D1F03V1ITEM, D1F03V1WARE, D1F03V1DIST, D1F03V1CSTI, D1F03V1NORA, D1F03V1ORL, D1F03V1STK, D1F03V1CST, D1F03V1ORDI, D1F03V1ORD, D1F03V1HIST, D1F03V1NORB
hdisk67	fcs7	

hdisk68	fcs6	D1F03V2ITEM, D1F03V2WARE, D1F03V2DIST, D1F03V2CSTI, D1F03V2NORA, D1F03V2ORL, D1F03V2STK, D1F03V2CST, D1F03V2ORDI, D1F03V2ORD, D1F03V2HIST, D1F03V2NORB
hdisk69	fcs7	
hdisk70	fcs6	D1F03V3ITEM, D1F03V3WARE, D1F03V3DIST, D1F03V3CSTI, D1F03V3NORA, D1F03V3ORL, D1F03V3STK, D1F03V3CST, D1F03V3ORDI, D1F03V3ORD, D1F03V3HIST, D1F03V3NORB
hdisk71	fcs7	
hdisk72	fcs6	D1F03V4ITEM, D1F03V4WARE, D1F03V4DIST, D1F03V4CSTI, D1F03V4NORA, D1F03V4ORL, D1F03V4STK, D1F03V4CST, D1F03V4ORDI, D1F03V4ORD, D1F03V4HIST, D1F03V4NORB
hdisk73	fcs7	
hdisk58	fcs8	D1F04V1ITEM, D1F04V1WARE, D1F04V1DIST, D1F04V1CSTI, D1F04V1NORA, D1F04V1ORL, D1F04V1STK, D1F04V1CST, D1F04V1ORDI, D1F04V1ORD, D1F04V1HIST, D1F04V1NORB
hdisk59	fcs5	
hdisk60	fcs8	D1F04V2ITEM, D1F04V2WARE, D1F04V2DIST, D1F04V2CSTI, D1F04V2NORA, D1F04V2ORL, D1F04V2STK, D1F04V2CST, D1F04V2ORDI, D1F04V2ORD, D1F04V2HIST, D1F04V2NORB
hdisk61	fcs5	
hdisk62	fcs8	D1F04V3ITEM, D1F04V3WARE, D1F04V3DIST, D1F04V3CSTI, D1F04V3NORA, D1F04V3ORL, D1F04V3STK, D1F04V3CST, D1F04V3ORDI, D1F04V3ORD, D1F04V3HIST, D1F04V3NORB
hdisk63	fcs5	
hdisk64	fcs8	D1F04V4ITEM, D1F04V4WARE, D1F04V4DIST, D1F04V4CSTI, D1F04V4NORA, D1F04V4ORL, D1F04V4STK, D1F04V4CST, D1F04V4ORDI, D1F04V4ORD, D1F04V4HIST, D1F04V4NORB
hdisk65	fcs5	
hdisk74	fcs9	D1F05V1ITEM, D1F05V1WARE, D1F05V1DIST, D1F05V1CSTI, D1F05V1NORA, D1F05V1ORL, D1F05V1STK, D1F05V1CST, D1F05V1ORDI, D1F05V1ORD, D1F05V1HIST, D1F05V1NORB
hdisk75	fcs10	
hdisk76	fcs9	D1F05V2ITEM, D1F05V2WARE, D1F05V2DIST, D1F05V2CSTI, D1F05V2NORA, D1F05V2ORL, D1F05V2STK, D1F05V2CST, D1F05V2ORDI, D1F05V2ORD, D1F05V2HIST, D1F05V2NORB
hdisk77	fcs10	
hdisk78	fcs9	D1F05V3ITEM, D1F05V3WARE, D1F05V3DIST, D1F05V3CSTI, D1F05V3NORA, D1F05V3ORL, D1F05V3STK, D1F05V3CST, D1F05V3ORDI, D1F05V3ORD, D1F05V3HIST, D1F05V3NORB
hdisk79	fcs10	
hdisk80	fcs9	D1F05V4ITEM, D1F05V4WARE, D1F05V4DIST, D1F05V4CSTI, D1F05V4NORA, D1F05V4ORL, D1F05V4STK, D1F05V4CST, D1F05V4ORDI, D1F05V4ORD, D1F05V4HIST, D1F05V4NORB
hdisk81	fcs10	
hdisk82	fcs14	D1F06V1ITEM, D1F06V1WARE, D1F06V1DIST, D1F06V1CSTI, D1F06V1NORA, D1F06V1ORL, D1F06V1STK, D1F06V1CST, D1F06V1ORDI, D1F06V1ORD, D1F06V1HIST, D1F06V1NORB
hdisk83	fcs11	
hdisk84	fcs14	D1F06V2ITEM, D1F06V2WARE, D1F06V2DIST, D1F06V2CSTI, D1F06V2NORA, D1F06V2ORL, D1F06V2STK, D1F06V2CST, D1F06V2ORDI, D1F06V2ORD, D1F06V2HIST, D1F06V2NORB
hdisk85	fcs11	
hdisk86	fcs14	D1F06V3ITEM, D1F06V3WARE, D1F06V3DIST, D1F06V3CSTI, D1F06V3NORA, D1F06V3ORL, D1F06V3STK, D1F06V3CST, D1F06V3ORDI, D1F06V3ORD, D1F06V3HIST, D1F06V3NORB
hdisk87	fcs11	
hdisk88	fcs14	D1F06V4ITEM, D1F06V4WARE, D1F06V4DIST, D1F06V4CSTI, D1F06V4NORA, D1F06V4ORL, D1F06V4STK, D1F06V4CST, D1F06V4ORDI, D1F06V4ORD, D1F06V4HIST, D1F06V4NORB
hdisk89	fcs11	
hdisk106	fcs18	D1F07V1ITEM, D1F07V1WARE, D1F07V1DIST, D1F07V1CSTI, D1F07V1NORA, D1F07V1ORL, D1F07V1STK, D1F07V1CST, D1F07V1ORDI, D1F07V1ORD, D1F07V1HIST, D1F07V1NORB
hdisk107	fcs17	
hdisk108	fcs18	D1F07V2ITEM, D1F07V2WARE, D1F07V2DIST, D1F07V2CSTI, D1F07V2NORA, D1F07V2ORL, D1F07V2STK, D1F07V2CST, D1F07V2ORDI, D1F07V2ORD, D1F07V2HIST, D1F07V2NORB
hdisk109	fcs17	
hdisk110	fcs18	D1F07V3ITEM, D1F07V3WARE, D1F07V3DIST, D1F07V3CSTI, D1F07V3NORA, D1F07V3ORL, D1F07V3STK, D1F07V3CST, D1F07V3ORDI, D1F07V3ORD, D1F07V3HIST, D1F07V3NORB
hdisk111	fcs17	
hdisk112	fcs18	D1F07V4ITEM, D1F07V4WARE, D1F07V4DIST, D1F07V4CSTI, D1F07V4NORA,

hdisk113	fcs17	D1F07V4ORL, D1F07V4STK, D1F07V4CST, D1F07V4ORDI, D1F07V4ORD, D1F07V4HIST, D1F07V4NORB
hdisk98	fcs19	D1F08V1ITEM, D1F08V1WARE, D1F08V1DIST, D1F08V1CSTI, D1F08V1NORA, D1F08V1ORL, D1F08V1STK, D1F08V1CST, D1F08V1ORDI, D1F08V1ORD, D1F08V1HIST, D1F08V1NORB
hdisk99	fcs16	
hdisk100	fcs19	D1F08V2ITEM, D1F08V2WARE, D1F08V2DIST, D1F08V2CSTI, D1F08V2NORA, D1F08V2ORL, D1F08V2STK, D1F08V2CST, D1F08V2ORDI, D1F08V2ORD, D1F08V2HIST, D1F08V2NORB
hdisk101	fcs16	
hdisk102	fcs19	D1F08V3ITEM, D1F08V3WARE, D1F08V3DIST, D1F08V3CSTI, D1F08V3NORA, D1F08V3ORL, D1F08V3STK, D1F08V3CST, D1F08V3ORDI, D1F08V3ORD, D1F08V3HIST, D1F08V3NORB
hdisk103	fcs16	
hdisk104	fcs19	D1F08V4ITEM, D1F08V4WARE, D1F08V4DIST, D1F08V4CSTI, D1F08V4NORA, D1F08V4ORL, D1F08V4STK, D1F08V4CST, D1F08V4ORDI, D1F08V4ORD, D1F08V4HIST, D1F08V4NORB
hdisk105	fcs16	
hdisk90	fcs12	D1F09V1ITEM, D1F09V1WARE, D1F09V1DIST, D1F09V1CSTI, D1F09V1NORA, D1F09V1ORL, D1F09V1STK, D1F09V1CST, D1F09V1ORDI, D1F09V1ORD, D1F09V1HIST, D1F09V1NORB
hdisk91	fcs13	
hdisk92	fcs12	D1F09V2ITEM, D1F09V2WARE, D1F09V2DIST, D1F09V2CSTI, D1F09V2NORA, D1F09V2ORL, D1F09V2STK, D1F09V2CST, D1F09V2ORDI, D1F09V2ORD, D1F09V2HIST, D1F09V2NORB
hdisk93	fcs13	
hdisk94	fcs12	D1F09V3ITEM, D1F09V3WARE, D1F09V3DIST, D1F09V3CSTI, D1F09V3NORA, D1F09V3ORL, D1F09V3STK, D1F09V3CST, D1F09V3ORDI, D1F09V3ORD, D1F09V3HIST, D1F09V3NORB
hdisk95	fcs13	
hdisk96	fcs13	D1F09V4ITEM, D1F09V4WARE, D1F09V4DIST, D1F09V4CSTI, D1F09V4NORA, D1F09V4ORL, D1F09V4STK, D1F09V4CST, D1F09V4ORDI, D1F09V4ORD, D1F09V4HIST, D1F09V4NORB
hdisk97	fcs12	
hdisk114	fcs20	D1F10V1ITEM, D1F10V1WARE, D1F10V1DIST, D1F10V1CSTI, D1F10V1NORA, D1F10V1ORL, D1F10V1STK, D1F10V1CST, D1F10V1ORDI, D1F10V1ORD, D1F10V1HIST, D1F10V1NORB
hdisk115	fcs21	
hdisk116	fcs20	D1F10V2ITEM, D1F10V2WARE, D1F10V2DIST, D1F10V2CSTI, D1F10V2NORA, D1F10V2ORL, D1F10V2STK, D1F10V2CST, D1F10V2ORDI, D1F10V2ORD, D1F10V2HIST, D1F10V2NORB
hdisk117	fcs21	
hdisk118	fcs20	D1F10V3ITEM, D1F10V3WARE, D1F10V3DIST, D1F10V3CSTI, D1F10V3NORA, D1F10V3ORL, D1F10V3STK, D1F10V3CST, D1F10V3ORDI, D1F10V3ORD, D1F10V3HIST, D1F10V3NORB
hdisk119	fcs21	
hdisk120	fcs20	D1F10V4ITEM, D1F10V4WARE, D1F10V4DIST, D1F10V4CSTI, D1F10V4NORA, D1F10V4ORL, D1F10V4STK, D1F10V4CST, D1F10V4ORDI, D1F10V4ORD, D1F10V4HIST, D1F10V4NORB
hdisk121	fcs21	
hdisk152	fcs29	D1F11V1ITEM, D1F11V1WARE, D1F11V1DIST, D1F11V1CSTI, D1F11V1NORA, D1F11V1ORL, D1F11V1STK, D1F11V1CST, D1F11V1ORDI, D1F11V1ORD, D1F11V1HIST, D1F11V1NORB
hdisk153	fcs30	
hdisk154	fcs29	D1F11V2ITEM, D1F11V2WARE, D1F11V2DIST, D1F11V2CSTI, D1F11V2NORA, D1F11V2ORL, D1F11V2STK, D1F11V2CST, D1F11V2ORDI, D1F11V2ORD, D1F11V2HIST, D1F11V2NORB
hdisk155	fcs30	
hdisk156	fcs29	D1F11V3ITEM, D1F11V3WARE, D1F11V3DIST, D1F11V3CSTI, D1F11V3NORA, D1F11V3ORL, D1F11V3STK, D1F11V3CST, D1F11V3ORDI, D1F11V3ORD, D1F11V3HIST, D1F11V3NORB
hdisk157	fcs30	
hdisk158	fcs29	D1F11V4ITEM, D1F11V4WARE, D1F11V4DIST, D1F11V4CSTI, D1F11V4NORA, D1F11V4ORL, D1F11V4STK, D1F11V4CST, D1F11V4ORDI, D1F11V4ORD, D1F11V4HIST, D1F11V4NORB
hdisk159	fcs30	
hdisk144	fcs28	D1F12V1ITEM, D1F12V1WARE, D1F12V1DIST, D1F12V1CSTI, D1F12V1NORA, D1F12V1ORL, D1F12V1STK, D1F12V1CST, D1F12V1ORDI, D1F12V1ORD, D1F12V1HIST, D1F12V1NORB
hdisk145	fcs31	
hdisk146	fcs28	D1F12V2ITEM, D1F12V2WARE, D1F12V2DIST, D1F12V2CSTI, D1F12V2NORA, D1F12V2ORL, D1F12V2STK, D1F12V2CST, D1F12V2ORDI, D1F12V2ORD, D1F12V2HIST, D1F12V2NORB
hdisk147	fcs31	

hdisk148	fcs28	D1F12V3ITEM, D1F12V3WARE, D1F12V3DIST, D1F12V3CSTI, D1F12V3NORA, D1F12V3ORL, D1F12V3STK, D1F12V3CST, D1F12V3ORDI, D1F12V3ORD, D1F12V3HIST, D1F12V3NORB
hdisk149	fcs31	
hdisk150	fcs28	D1F12V4ITEM, D1F12V4WARE, D1F12V4DIST, D1F12V4CSTI, D1F12V4NORA, D1F12V4ORL, D1F12V4STK, D1F12V4CST, D1F12V4ORDI, D1F12V4ORD, D1F12V4HIST, D1F12V4NORB
hdisk151	fcs31	
hdisk130	fcs23	D1F12V1ITEM, D1F12V1WARE, D1F12V1DIST, D1F12V1CSTI, D1F12V1NORA, D1F12V1ORL, D1F12V1STK, D1F12V1CST, D1F12V1ORDI, D1F12V1ORD, D1F12V1HIST, D1F12V1NORB
hdisk131	fcs24	
hdisk132	fcs23	D1F12V2ITEM, D1F12V2WARE, D1F12V2DIST, D1F12V2CSTI, D1F12V2NORA, D1F12V2ORL, D1F12V2STK, D1F12V2CST, D1F12V2ORDI, D1F12V2ORD, D1F12V2HIST, D1F12V2NORB
hdisk133	fcs24	
hdisk134	fcs23	D1F12V3ITEM, D1F12V3WARE, D1F12V3DIST, D1F12V3CSTI, D1F12V3NORA, D1F12V3ORL, D1F12V3STK, D1F12V3CST, D1F12V3ORDI, D1F12V3ORD, D1F12V3HIST, D1F12V3NORB
hdisk135	fcs24	
hdisk136	fcs23	D1F12V4ITEM, D1F12V4WARE, D1F12V4DIST, D1F12V4CSTI, D1F12V4NORA, D1F12V4ORL, D1F12V4STK, D1F12V4CST, D1F12V4ORDI, D1F12V4ORD, D1F12V4HIST, D1F12V4NORB
hdisk137	fcs24	
hdisk122	fcs22	D1F13V1ITEM, D1F13V1WARE, D1F13V1DIST, D1F13V1CSTI, D1F13V1NORA, D1F13V1ORL, D1F13V1STK, D1F13V1CST, D1F13V1ORDI, D1F13V1ORD, D1F13V1HIST, D1F13V1NORB
hdisk123	fcs26	
hdisk124	fcs22	D1F13V2ITEM, D1F13V2WARE, D1F13V2DIST, D1F13V2CSTI, D1F13V2NORA, D1F13V2ORL, D1F13V2STK, D1F13V2CST, D1F13V2ORDI, D1F13V2ORD, D1F13V2HIST, D1F13V2NORB
hdisk125	fcs26	
hdisk126	fcs22	D1F13V3ITEM, D1F13V3WARE, D1F13V3DIST, D1F13V3CSTI, D1F13V3NORA, D1F13V3ORL, D1F13V3STK, D1F13V3CST, D1F13V3ORDI, D1F13V3ORD, D1F13V3HIST, D1F13V3NORB
hdisk127	fcs26	
hdisk128	fcs22	D1F13V4ITEM, D1F13V4WARE, D1F13V4DIST, D1F13V4CSTI, D1F13V4NORA, D1F13V4ORL, D1F13V4STK, D1F13V4CST, D1F13V4ORDI, D1F13V4ORD, D1F13V4HIST, D1F13V4NORB
hdisk129	fcs26	
hdisk160	fcs32	D1F14V1ITEM, D1F14V1WARE, D1F14V1DIST, D1F14V1CSTI, D1F14V1NORA, D1F14V1ORL, D1F14V1STK, D1F14V1CST, D1F14V1ORDI, D1F14V1ORD, D1F14V1HIST, D1F14V1NORB
hdisk161	fcs33	
hdisk162	fcs32	D1F14V2ITEM, D1F14V2WARE, D1F14V2DIST, D1F14V2CSTI, D1F14V2NORA, D1F14V2ORL, D1F14V2STK, D1F14V2CST, D1F14V2ORDI, D1F14V2ORD, D1F14V2HIST, D1F14V2NORB
hdisk163	fcs33	
hdisk164	fcs32	D1F14V3ITEM, D1F14V3WARE, D1F14V3DIST, D1F14V3CSTI, D1F14V3NORA, D1F14V3ORL, D1F14V3STK, D1F14V3CST, D1F14V3ORDI, D1F14V3ORD, D1F14V3HIST, D1F14V3NORB
hdisk165	fcs33	
hdisk166	fcs32	D1F14V4ITEM, D1F14V4WARE, D1F14V4DIST, D1F14V4CSTI, D1F14V4NORA, D1F14V4ORL, D1F14V4STK, D1F14V4CST, D1F14V4ORDI, D1F14V4ORD, D1F14V4HIST, D1F14V4NORB
hdisk167	fcs33	
hdisk168	fcs37	D1F15V1ITEM, D1F15V1WARE, D1F15V1DIST, D1F15V1CSTI, D1F15V1NORA, D1F15V1ORL, D1F15V1STK, D1F15V1CST, D1F15V1ORDI, D1F15V1ORD, D1F15V1HIST, D1F15V1NORB
hdisk169	fcs34	
hdisk170	fcs37	D1F15V2ITEM, D1F15V2WARE, D1F15V2DIST, D1F15V2CSTI, D1F15V2NORA, D1F15V2ORL, D1F15V2STK, D1F15V2CST, D1F15V2ORDI, D1F15V2ORD, D1F15V2HIST, D1F15V2NORB
hdisk171	fcs34	
hdisk172	fcs37	D1F15V3ITEM, D1F15V3WARE, D1F15V3DIST, D1F15V3CSTI, D1F15V3NORA, D1F15V3ORL, D1F15V3STK, D1F15V3CST, D1F15V3ORDI, D1F15V3ORD, D1F15V3HIST, D1F15V3NORB
hdisk173	fcs34	
hdisk174	fcs37	D1F15V4ITEM, D1F15V4WARE, D1F15V4DIST, D1F15V4CSTI, D1F15V4NORA, D1F15V4ORL, D1F15V4STK, D1F15V4CST, D1F15V4ORDI, D1F15V4ORD, D1F15V4HIST, D1F15V4NORB
hdisk175	fcs34	
hdisk200	fcs41	D1F16V1ITEM, D1F16V1WARE, D1F16V1DIST, D1F16V1CSTI, D1F16V1NORA,

hdisk201	fcs42	D1F16V1ORL, D1F16V1STK, D1F16V1CST, D1F16V1ORDI, D1F16V1ORD, D1F16V1HIST, D1F16V1NORB
hdisk202	fcs41	D1F16V2ITEM, D1F16V2WARE, D1F16V2DIST, D1F16V2CSTI, D1F16V2NORA, D1F16V2ORL, D1F16V2STK, D1F16V2CST, D1F16V2ORDI, D1F16V2ORD, D1F16V2HIST, D1F16V2NORB
hdisk203	fcs42	
hdisk204	fcs41	D1F16V3ITEM, D1F16V3WARE, D1F16V3DIST, D1F16V3CSTI, D1F16V3NORA, D1F16V3ORL, D1F16V3STK, D1F16V3CST, D1F16V3ORDI, D1F16V3ORD, D1F16V3HIST, D1F16V3NORB
hdisk205	fcs42	
hdisk206	fcs41	D1F16V4ITEM, D1F16V4WARE, D1F16V4DIST, D1F16V4CSTI, D1F16V4NORA, D1F16V4ORL, D1F16V4STK, D1F16V4CST, D1F16V4ORDI, D1F16V4ORD, D1F16V4HIST, D1F16V4NORB
hdisk207	fcs42	
hdisk192	fcs44	D1F17V1ITEM, D1F17V1WARE, D1F17V1DIST, D1F17V1CSTI, D1F17V1NORA, D1F17V1ORL, D1F17V1STK, D1F17V1CST, D1F17V1ORDI, D1F17V1ORD, D1F17V1HIST, D1F17V1NORB
hdisk193	fcs40	
hdisk194	fcs44	D1F17V2ITEM, D1F17V2WARE, D1F17V2DIST, D1F17V2CSTI, D1F17V2NORA, D1F17V2ORL, D1F17V2STK, D1F17V2CST, D1F17V2ORDI, D1F17V2ORD, D1F17V2HIST, D1F17V2NORB
hdisk195	fcs40	
hdisk196	fcs44	D1F17V3ITEM, D1F17V3WARE, D1F17V3DIST, D1F17V3CSTI, D1F17V3NORA, D1F17V3ORL, D1F17V3STK, D1F17V3CST, D1F17V3ORDI, D1F17V3ORD, D1F17V3HIST, D1F17V3NORB
hdisk197	fcs40	
hdisk198	fcs44	D1F17V4ITEM, D1F17V4WARE, D1F17V4DIST, D1F17V4CSTI, D1F17V4NORA, D1F17V4ORL, D1F17V4STK, D1F17V4CST, D1F17V4ORDI, D1F17V4ORD, D1F17V4HIST, D1F17V4NORB
hdisk199	fcs40	
hdisk176	fcs35	D1F18V1ITEM, D1F18V1WARE, D1F18V1DIST, D1F18V1CSTI, D1F18V1NORA, D1F18V1ORL, D1F18V1STK, D1F18V1CST, D1F18V1ORDI, D1F18V1ORD, D1F18V1HIST, D1F18V1NORB
hdisk177	fcs36	
hdisk178	fcs35	D1F18V2ITEM, D1F18V2WARE, D1F18V2DIST, D1F18V2CSTI, D1F18V2NORA, D1F18V2ORL, D1F18V2STK, D1F18V2CST, D1F18V2ORDI, D1F18V2ORD, D1F18V2HIST, D1F18V2NORB
hdisk179	fcs36	
hdisk180	fcs35	D1F18V3ITEM, D1F18V3WARE, D1F18V3DIST, D1F18V3CSTI, D1F18V3NORA, D1F18V3ORL, D1F18V3STK, D1F18V3CST, D1F18V3ORDI, D1F18V3ORD, D1F18V3HIST, D1F18V3NORB
hdisk181	fcs36	
hdisk182	fcs35	D1F18V4ITEM, D1F18V4WARE, D1F18V4DIST, D1F18V4CSTI, D1F18V4NORA, D1F18V4ORL, D1F18V4STK, D1F18V4CST, D1F18V4ORDI, D1F18V4ORD, D1F18V4HIST, D1F18V4NORB
hdisk183	fcs36	
hdisk184	fcs38	D1F19V1ITEM, D1F19V1WARE, D1F19V1DIST, D1F19V1CSTI, D1F19V1NORA, D1F19V1ORL, D1F19V1STK, D1F19V1CST, D1F19V1ORDI, D1F19V1ORD, D1F19V1HIST, D1F19V1NORB
hdisk185	fcs39	
hdisk186	fcs38	D1F19V2ITEM, D1F19V2WARE, D1F19V2DIST, D1F19V2CSTI, D1F19V2NORA, D1F19V2ORL, D1F19V2STK, D1F19V2CST, D1F19V2ORDI, D1F19V2ORD, D1F19V2HIST, D1F19V2NORB
hdisk187	fcs39	
hdisk188	fcs38	D1F19V3ITEM, D1F19V3WARE, D1F19V3DIST, D1F19V3CSTI, D1F19V3NORA, D1F19V3ORL, D1F19V3STK, D1F19V3CST, D1F19V3ORDI, D1F19V3ORD, D1F19V3HIST, D1F19V3NORB
hdisk189	fcs39	
hdisk190	fcs38	D1F19V4ITEM, D1F19V4WARE, D1F19V4DIST, D1F19V4CSTI, D1F19V4NORA, D1F19V4ORL, D1F19V4STK, D1F19V4CST, D1F19V4ORDI, D1F19V4ORD, D1F19V4HIST, D1F19V4NORB
hdisk191	fcs39	
hdisk341	fcs80	D1F20V1ITEM, D1F20V1WARE, D1F20V1DIST, D1F20V1CSTI, D1F20V1NORA, D1F20V1ORL, D1F20V1STK, D1F20V1CST, D1F20V1ORDI, D1F20V1ORD, D1F20V1HIST, D1F20V1NORB
hdisk342	fcs81	
hdisk343	fcs80	D1F20V2ITEM, D1F20V2WARE, D1F20V2DIST, D1F20V2CSTI, D1F20V2NORA, D1F20V2ORL, D1F20V2STK, D1F20V2CST, D1F20V2ORDI, D1F20V2ORD, D1F20V2HIST, D1F20V2NORB
hdisk344	fcs81	
hdisk345	fcs80	D1F20V3ITEM, D1F20V3WARE, D1F20V3DIST, D1F20V3CSTI, D1F20V3NORA, D1F20V3ORL, D1F20V3STK, D1F20V3CST, D1F20V3ORDI, D1F20V3ORD, D1F20V3HIST, D1F20V3NORB
hdisk346	fcs81	

hdisk347	fcs80	D1F20V4ITEM, D1F20V4WARE, D1F20V4DIST, D1F20V4CSTI, D1F20V4NORA, D1F20V4ORL, D1F20V4STK, D1F20V4CST, D1F20V4ORDI, D1F20V4ORD, D1F20V4HIST, D1F20V4NORB
hdisk348	fcs81	
hdisk333	fcs82	D1F21V1ITEM, D1F21V1WARE, D1F21V1DIST, D1F21V1CSTI, D1F21V1NORA, D1F21V1ORL, D1F21V1STK, D1F21V1CST, D1F21V1ORDI, D1F21V1ORD, D1F21V1HIST, D1F21V1NORB
hdisk334	fcs79	
hdisk335	fcs82	D1F21V2ITEM, D1F21V2WARE, D1F21V2DIST, D1F21V2CSTI, D1F21V2NORA, D1F21V2ORL, D1F21V2STK, D1F21V2CST, D1F21V2ORDI, D1F21V2ORD, D1F21V2HIST, D1F21V2NORB
hdisk336	fcs79	
hdisk337	fcs82	D1F21V3ITEM, D1F21V3WARE, D1F21V3DIST, D1F21V3CSTI, D1F21V3NORA, D1F21V3ORL, D1F21V3STK, D1F21V3CST, D1F21V3ORDI, D1F21V3ORD, D1F21V3HIST, D1F21V3NORB
hdisk338	fcs79	
hdisk339	fcs82	D1F21V4ITEM, D1F21V4WARE, D1F21V4DIST, D1F21V4CSTI, D1F21V4NORA, D1F21V4ORL, D1F21V4STK, D1F21V4CST, D1F21V4ORDI, D1F21V4ORD, D1F21V4HIST, D1F21V4NORB
hdisk340	fcs79	
hdisk357	fcs85	D1F22V1ITEM, D1F22V1WARE, D1F22V1DIST, D1F22V1CSTI, D1F22V1NORA, D1F22V1ORL, D1F22V1STK, D1F22V1CST, D1F22V1ORDI, D1F22V1ORD, D1F22V1HIST, D1F22V1NORB
hdisk358	fcs86	
hdisk359	fcs85	D1F22V2ITEM, D1F22V2WARE, D1F22V2DIST, D1F22V2CSTI, D1F22V2NORA, D1F22V2ORL, D1F22V2STK, D1F22V2CST, D1F22V2ORDI, D1F22V2ORD, D1F22V2HIST, D1F22V2NORB
hdisk360	fcs86	
hdisk361	fcs85	D1F22V3ITEM, D1F22V3WARE, D1F22V3DIST, D1F22V3CSTI, D1F22V3NORA, D1F22V3ORL, D1F22V3STK, D1F22V3CST, D1F22V3ORDI, D1F22V3ORD, D1F22V3HIST, D1F22V3NORB
hdisk362	fcs86	
hdisk363	fcs85	D1F22V4ITEM, D1F22V4WARE, D1F22V4DIST, D1F22V4CSTI, D1F22V4NORA, D1F22V4ORL, D1F22V4STK, D1F22V4CST, D1F22V4ORDI, D1F22V4ORD, D1F22V4HIST, D1F22V4NORB
hdisk364	fcs86	
hdisk349	fcs84	D1F23V1ITEM, D1F23V1WARE, D1F23V1DIST, D1F23V1CSTI, D1F23V1NORA, D1F23V1ORL, D1F23V1STK, D1F23V1CST, D1F23V1ORDI, D1F23V1ORD, D1F23V1HIST, D1F23V1NORB
hdisk350	fcs87	
hdisk351	fcs84	D1F23V2ITEM, D1F23V2WARE, D1F23V2DIST, D1F23V2CSTI, D1F23V2NORA, D1F23V2ORL, D1F23V2STK, D1F23V2CST, D1F23V2ORDI, D1F23V2ORD, D1F23V2HIST, D1F23V2NORB
hdisk352	fcs87	
hdisk353	fcs84	D1F23V3ITEM, D1F23V3WARE, D1F23V3DIST, D1F23V3CSTI, D1F23V3NORA, D1F23V3ORL, D1F23V3STK, D1F23V3CST, D1F23V3ORDI, D1F23V3ORD, D1F23V3HIST, D1F23V3NORB
hdisk354	fcs87	
hdisk355	fcs84	D1F23V4ITEM, D1F23V4WARE, D1F23V4DIST, D1F23V4CSTI, D1F23V4NORA, D1F23V4ORL, D1F23V4STK, D1F23V4CST, D1F23V4ORDI, D1F23V4ORD, D1F23V4HIST, D1F23V4NORB
hdisk356	fcs87	
hdisk365	fcs88	D1F24V1ITEM, D1F24V1WARE, D1F24V1DIST, D1F24V1CSTI, D1F24V1NORA, D1F24V1ORL, D1F24V1STK, D1F24V1CST, D1F24V1ORDI, D1F24V1ORD, D1F24V1HIST, D1F24V1NORB
hdisk366	fcs89	
hdisk367	fcs88	D1F24V2ITEM, D1F24V2WARE, D1F24V2DIST, D1F24V2CSTI, D1F24V2NORA, D1F24V2ORL, D1F24V2STK, D1F24V2CST, D1F24V2ORDI, D1F24V2ORD, D1F24V2HIST, D1F24V2NORB
hdisk368	fcs89	
hdisk369	fcs88	D1F24V3ITEM, D1F24V3WARE, D1F24V3DIST, D1F24V3CSTI, D1F24V3NORA, D1F24V3ORL, D1F24V3STK, D1F24V3CST, D1F24V3ORDI, D1F24V3ORD, D1F24V3HIST, D1F24V3NORB
hdisk370	fcs89	
hdisk371	fcs88	D1F24V4ITEM, D1F24V4WARE, D1F24V4DIST, D1F24V4CSTI, D1F24V4NORA, D1F24V4ORL, D1F24V4STK, D1F24V4CST, D1F24V4ORDI, D1F24V4ORD, D1F24V4HIST, D1F24V4NORB
hdisk372	fcs89	
hdisk298	fcs69	D1F25V1ITEM, D1F25V1WARE, D1F25V1DIST, D1F25V1CSTI, D1F25V1NORA, D1F25V1ORL, D1F25V1STK, D1F25V1CST, D1F25V1ORDI, D1F25V1ORD, D1F25V1HIST, D1F25V1NORB
hdisk299	fcs70	
hdisk300	fcs69	D1F25V2ITEM, D1F25V2WARE, D1F25V2DIST, D1F25V2CSTI, D1F25V2NORA,

hdisk258	fcs58	D1F30V1ITEM, D1F30V1WARE, D1F30V1DIST, D1F30V1CSTI, D1F30V1NORA, D1F30V1ORL, D1F30V1STK, D1F30V1CST, D1F30V1ORDI, D1F30V1ORD, D1F30V1HIST, D1F30V1NORB
hdisk259	fcs59	
hdisk260	fcs58	D1F30V2ITEM, D1F30V2WARE, D1F30V2DIST, D1F30V2CSTI, D1F30V2NORA, D1F30V2ORL, D1F30V2STK, D1F30V2CST, D1F30V2ORDI, D1F30V2ORD, D1F30V2HIST, D1F30V2NORB
hdisk261	fcs59	
hdisk262	fcs58	D1F30V3ITEM, D1F30V3WARE, D1F30V3DIST, D1F30V3CSTI, D1F30V3NORA, D1F30V3ORL, D1F30V3STK, D1F30V3CST, D1F30V3ORDI, D1F30V3ORD, D1F30V3HIST, D1F30V3NORB
hdisk263	fcs59	
hdisk264	fcs58	D1F30V4ITEM, D1F30V4WARE, D1F30V4DIST, D1F30V4CSTI, D1F30V4NORA, D1F30V4ORL, D1F30V4STK, D1F30V4CST, D1F30V4ORDI, D1F30V4ORD, D1F30V4HIST, D1F30V4NORB
hdisk265	fcs59	
hdisk250	fcs60	D1F31V1ITEM, D1F31V1WARE, D1F31V1DIST, D1F31V1CSTI, D1F31V1NORA, D1F31V1ORL, D1F31V1STK, D1F31V1CST, D1F31V1ORDI, D1F31V1ORD, D1F31V1HIST, D1F31V1NORB
hdisk251	fcs57	
hdisk252	fcs60	D1F31V2ITEM, D1F31V2WARE, D1F31V2DIST, D1F31V2CSTI, D1F31V2NORA, D1F31V2ORL, D1F31V2STK, D1F31V2CST, D1F31V2ORDI, D1F31V2ORD, D1F31V2HIST, D1F31V2NORB
hdisk253	fcs57	
hdisk254	fcs60	D1F31V3ITEM, D1F31V3WARE, D1F31V3DIST, D1F31V3CSTI, D1F31V3NORA, D1F31V3ORL, D1F31V3STK, D1F31V3CST, D1F31V3ORDI, D1F31V3ORD, D1F31V3HIST, D1F31V3NORB
hdisk255	fcs57	
hdisk256	fcs60	D1F31V4ITEM, D1F31V4WARE, D1F31V4DIST, D1F31V4CSTI, D1F31V4NORA, D1F31V4ORL, D1F31V4STK, D1F31V4CST, D1F31V4ORDI, D1F31V4ORD, D1F31V4HIST, D1F31V4NORB
hdisk257	fcs57	
hdisk274	fcs64	D1F32V1ITEM, D1F32V1WARE, D1F32V1DIST, D1F32V1CSTI, D1F32V1NORA, D1F32V1ORL, D1F32V1STK, D1F32V1CST, D1F32V1ORDI, D1F32V1ORD, D1F32V1HIST, D1F32V1NORB
hdisk275	fcs63	
hdisk276	fcs64	D1F32V2ITEM, D1F32V2WARE, D1F32V2DIST, D1F32V2CSTI, D1F32V2NORA, D1F32V2ORL, D1F32V2STK, D1F32V2CST, D1F32V2ORDI, D1F32V2ORD, D1F32V2HIST, D1F32V2NORB
hdisk277	fcs63	
hdisk278	fcs64	D1F32V3ITEM, D1F32V3WARE, D1F32V3DIST, D1F32V3CSTI, D1F32V3NORA, D1F32V3ORL, D1F32V3STK, D1F32V3CST, D1F32V3ORDI, D1F32V3ORD, D1F32V3HIST, D1F32V3NORB
hdisk279	fcs63	
hdisk280	fcs64	D1F32V4ITEM, D1F32V4WARE, D1F32V4DIST, D1F32V4CSTI, D1F32V4NORA, D1F32V4ORL, D1F32V4STK, D1F32V4CST, D1F32V4ORDI, D1F32V4ORD, D1F32V4HIST, D1F32V4NORB
hdisk281	fcs63	
hdisk266	fcs65	D1F33V1ITEM, D1F33V1WARE, D1F33V1DIST, D1F33V1CSTI, D1F33V1NORA, D1F33V1ORL, D1F33V1STK, D1F33V1CST, D1F33V1ORDI, D1F33V1ORD, D1F33V1HIST, D1F33V1NORB
hdisk267	fcs62	
hdisk268	fcs65	D1F33V2ITEM, D1F33V2WARE, D1F33V2DIST, D1F33V2CSTI, D1F33V2NORA, D1F33V2ORL, D1F33V2STK, D1F33V2CST, D1F33V2ORDI, D1F33V2ORD, D1F33V2HIST, D1F33V2NORB
hdisk269	fcs62	
hdisk270	fcs65	D1F33V3ITEM, D1F33V3WARE, D1F33V3DIST, D1F33V3CSTI, D1F33V3NORA, D1F33V3ORL, D1F33V3STK, D1F33V3CST, D1F33V3ORDI, D1F33V3ORD, D1F33V3HIST, D1F33V3NORB
hdisk271	fcs62	
hdisk272	fcs65	D1F33V4ITEM, D1F33V4WARE, D1F33V4DIST, D1F33V4CSTI, D1F33V4NORA, D1F33V4ORL, D1F33V4STK, D1F33V4CST, D1F33V4ORDI, D1F33V4ORD, D1F33V4HIST, D1F33V4NORB
hdisk273	fcs62	
hdisk282	fcs66	D1F34V1ITEM, D1F34V1WARE, D1F34V1DIST, D1F34V1CSTI, D1F34V1NORA, D1F34V1ORL, D1F34V1STK, D1F34V1CST, D1F34V1ORDI, D1F34V1ORD, D1F34V1HIST, D1F34V1NORB
hdisk283	fcs67	
hdisk284	fcs66	D1F34V2ITEM, D1F34V2WARE, D1F34V2DIST, D1F34V2CSTI, D1F34V2NORA, D1F34V2ORL, D1F34V2STK, D1F34V2CST, D1F34V2ORDI, D1F34V2ORD, D1F34V2HIST, D1F34V2NORB
hdisk285	fcs67	
hdisk286	fcs66	D1F34V3ITEM, D1F34V3WARE, D1F34V3DIST, D1F34V3CSTI, D1F34V3NORA,

hdisk287	fcs67	D1F34V3ORL, D1F34V3STK, D1F34V3CST, D1F34V3ORDI, D1F34V3ORD, D1F34V3HIST, D1F34V3NORB
hdisk288	fcs66	D1F34V4ITEM, D1F34V4WARE, D1F34V4DIST, D1F34V4CSTI, D1F34V4NORA, D1F34V4ORL, D1F34V4STK, D1F34V4CST, D1F34V4ORDI, D1F34V4ORD, D1F34V4HIST, D1F34V4NORB
hdisk289	fcs67	
hdisk216	fcs48	D1F35V1ITEM, D1F35V1WARE, D1F35V1DIST, D1F35V1CSTI, D1F35V1NORA, D1F35V1ORL, D1F35V1STK, D1F35V1CST, D1F35V1ORDI, D1F35V1ORD, D1F35V1HIST, D1F35V1NORB
hdisk217	fcs47	
hdisk218	fcs48	D1F35V2ITEM, D1F35V2WARE, D1F35V2DIST, D1F35V2CSTI, D1F35V2NORA, D1F35V2ORL, D1F35V2STK, D1F35V2CST, D1F35V2ORDI, D1F35V2ORD, D1F35V2HIST, D1F35V2NORB
hdisk219	fcs47	
hdisk220	fcs48	D1F35V3ITEM, D1F35V3WARE, D1F35V3DIST, D1F35V3CSTI, D1F35V3NORA, D1F35V3ORL, D1F35V3STK, D1F35V3CST, D1F35V3ORDI, D1F35V3ORD, D1F35V3HIST, D1F35V3NORB
hdisk221	fcs47	
hdisk222	fcs48	D1F35V4ITEM, D1F35V4WARE, D1F35V4DIST, D1F35V4CSTI, D1F35V4NORA, D1F35V4ORL, D1F35V4STK, D1F35V4CST, D1F35V4ORDI, D1F35V4ORD, D1F35V4HIST, D1F35V4NORB
hdisk223	fcs47	
hdisk208	fcs49	D1F36V1ITEM, D1F36V1WARE, D1F36V1DIST, D1F36V1CSTI, D1F36V1NORA, D1F36V1ORL, D1F36V1STK, D1F36V1CST, D1F36V1ORDI, D1F36V1ORD, D1F36V1HIST, D1F36V1NORB
hdisk209	fcs46	
hdisk210	fcs49	D1F36V2ITEM, D1F36V2WARE, D1F36V2DIST, D1F36V2CSTI, D1F36V2NORA, D1F36V2ORL, D1F36V2STK, D1F36V2CST, D1F36V2ORDI, D1F36V2ORD, D1F36V2HIST, D1F36V2NORB
hdisk211	fcs46	
hdisk212	fcs49	D1F36V3ITEM, D1F36V3WARE, D1F36V3DIST, D1F36V3CSTI, D1F36V3NORA, D1F36V3ORL, D1F36V3STK, D1F36V3CST, D1F36V3ORDI, D1F36V3ORD, D1F36V3HIST, D1F36V3NORB
hdisk213	fcs46	
hdisk214	fcs49	D1F36V4ITEM, D1F36V4WARE, D1F36V4DIST, D1F36V4CSTI, D1F36V4NORA, D1F36V4ORL, D1F36V4STK, D1F36V4CST, D1F36V4ORDI, D1F36V4ORD, D1F36V4HIST, D1F36V4NORB
hdisk215	fcs46	
hdisk234	fcs52	D1F37V1ITEM, D1F37V1WARE, D1F37V1DIST, D1F37V1CSTI, D1F37V1NORA, D1F37V1ORL, D1F37V1STK, D1F37V1CST, D1F37V1ORDI, D1F37V1ORD, D1F37V1HIST, D1F37V1NORB
hdisk235	fcs53	
hdisk236	fcs52	D1F37V2ITEM, D1F37V2WARE, D1F37V2DIST, D1F37V2CSTI, D1F37V2NORA, D1F37V2ORL, D1F37V2STK, D1F37V2CST, D1F37V2ORDI, D1F37V2ORD, D1F37V2HIST, D1F37V2NORB
hdisk237	fcs53	
hdisk238	fcs52	D1F37V3ITEM, D1F37V3WARE, D1F37V3DIST, D1F37V3CSTI, D1F37V3NORA, D1F37V3ORL, D1F37V3STK, D1F37V3CST, D1F37V3ORDI, D1F37V3ORD, D1F37V3HIST, D1F37V3NORB
hdisk239	fcs53	
hdisk240	fcs52	D1F37V4ITEM, D1F37V4WARE, D1F37V4DIST, D1F37V4CSTI, D1F37V4NORA, D1F37V4ORL, D1F37V4STK, D1F37V4CST, D1F37V4ORDI, D1F37V4ORD, D1F37V4HIST, D1F37V4NORB
hdisk241	fcs53	
hdisk226	fcs54	D1F38V1ITEM, D1F38V1WARE, D1F38V1DIST, D1F38V1CSTI, D1F38V1NORA, D1F38V1ORL, D1F38V1STK, D1F38V1CST, D1F38V1ORDI, D1F38V1ORD, D1F38V1HIST, D1F38V1NORB
hdisk227	fcs51	
hdisk228	fcs54	D1F38V2ITEM, D1F38V2WARE, D1F38V2DIST, D1F38V2CSTI, D1F38V2NORA, D1F38V2ORL, D1F38V2STK, D1F38V2CST, D1F38V2ORDI, D1F38V2ORD, D1F38V2HIST, D1F38V2NORB
hdisk229	fcs51	
hdisk230	fcs54	D1F38V3ITEM, D1F38V3WARE, D1F38V3DIST, D1F38V3CSTI, D1F38V3NORA, D1F38V3ORL, D1F38V3STK, D1F38V3CST, D1F38V3ORDI, D1F38V3ORD, D1F38V3HIST, D1F38V3NORB
hdisk231	fcs51	
hdisk232	fcs54	D1F38V4ITEM, D1F38V4WARE, D1F38V4DIST, D1F38V4CSTI, D1F38V4NORA, D1F38V4ORL, D1F38V4STK, D1F38V4CST, D1F38V4ORDI, D1F38V4ORD, D1F38V4HIST, D1F38V4NORB
hdisk233	fcs51	
hdisk242	fcs55	D1F39V1ITEM, D1F39V1WARE, D1F39V1DIST, D1F39V1CSTI, D1F39V1NORA, D1F39V1ORL, D1F39V1STK, D1F39V1CST, D1F39V1ORDI, D1F39V1ORD, D1F39V1HIST, D1F39V1NORB
hdisk243	fcs56	

hdisk244	fcs55	D1F39V2ITEM, D1F39V2WARE, D1F39V2DIST, D1F39V2CSTI, D1F39V2NORA, D1F39V2ORL, D1F39V2STK, D1F39V2CST, D1F39V2ORDI, D1F39V2ORD, D1F39V2HIST, D1F39V2NORB
hdisk245	fcs56	
hdisk246	fcs55	D1F39V3ITEM, D1F39V3WARE, D1F39V3DIST, D1F39V3CSTI, D1F39V3NORA, D1F39V3ORL, D1F39V3STK, D1F39V3CST, D1F39V3ORDI, D1F39V3ORD, D1F39V3HIST, D1F39V3NORB
hdisk246	fcs55	
hdisk246	fcs55	D1F39V4ITEM, D1F39V4WARE, D1F39V4DIST, D1F39V4CSTI, D1F39V4NORA, D1F39V4ORL, D1F39V4STK, D1F39V4CST, D1F39V4ORDI, D1F39V4ORD, D1F39V4HIST, D1F39V4NORB
hdisk247	fcs56	
hdisk248	fcs55	D1F40V1ITEM, D1F40V1WARE, D1F40V1DIST, D1F40V1CSTI, D1F40V1NORA, D1F40V1ORL, D1F40V1STK, D1F40V1CST, D1F40V1ORDI, D1F40V1ORD, D1F40V1HIST, D1F40V1NORB
hdisk48	fcs4	
hdisk49	fcs1	D1F40V2ITEM, D1F40V2WARE, D1F40V2DIST, D1F40V2CSTI, D1F40V2NORA, D1F40V2ORL, D1F40V2STK, D1F40V2CST, D1F40V2ORDI, D1F40V2ORD, D1F40V2HIST, D1F40V2NORB
hdisk50	fcs4	
hdisk51	fcs1	D1F40V3ITEM, D1F40V3WARE, D1F40V3DIST, D1F40V3CSTI, D1F40V3NORA, D1F40V3ORL, D1F40V3STK, D1F40V3CST, D1F40V3ORDI, D1F40V3ORD, D1F40V3HIST, D1F40V3NORB
hdisk52	fcs4	
hdisk53	fcs1	D1F40V4ITEM, D1F40V4WARE, D1F40V4DIST, D1F40V4CSTI, D1F40V4NORA, D1F40V4ORL, D1F40V4STK, D1F40V4CST, D1F40V4ORDI, D1F40V4ORD, D1F40V4HIST, D1F40V4NORB
hdisk54	fcs4	

Table 4-2: IBM eServer p5 595 Model 9119-595 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	256,000				
Measured TpmC	3,210,540.63				
Table	Rows	Table	Index	5% Space	Total Space
Warehouse	256,000	60	0	3	63
District	2,560,000	294	0	15	309
Item	100,000	10	0	1	11
Stock	25,600,000,000	8,334,080	0	416,704	8,750,784
Customer	7,680,000,000	6,000,800	370,880	318,584	6,690,264
New-Order	2,304,000,000	178,672	0	0	178,672
Orders	7,680,000,000	290,400	216,000	0	506,400
Order-Line	115,200,000,000	7,724,000	0	0	7,724,000
History	7,680,000,000	472,768	0	0	472,768
Free Space	1,571,142		<u>30 Minute log Computations</u>		
Dynamic Space	8,487,168			Log Written (KB)	227,306,277
Static Space	15,836,102			Total New-Order Txns	96,316,219
Daily Growth	1,703,025			Log Written per New-Order (KB)	2.36
Daily Spread	0				
Data Storage Requirement					
60 Days (MB)	118,017,594				
60 Days (GB)	115,252				
Log Storage Requirement					
8 Hours (GB)	3,468				
Disk Sizing					
	Formatted	SUT		Priced	
Disk Type	Capacity (GB)	# of Disks	Capacity (GB)	# of Disks	Capacity (GB)
DB FastT 36.4GB	33.40	6,400	213,760	6,400	213,760
LOG FastT 13+P RAID5	882.26	10	8,823	10	8,823
OS SCSI 36GB	33.90	8	271	8	271
Total Capacity					222,854

5 Clause 5: Performance Metrics and Response Time Related Items

5.1. Response Times

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

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

5.2. Keying and Think Times

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

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

Response Times	New Order	Payment	Order Status	Delivery (int./def.)	Stock Level	Menus
90 %	0.53	0.53	0.53	0.27/0.20	0.53	0.25
Average	0.34	0.34	0.34	0.16/0.15	0.35	0.16
Maximum	6.84	7.17	6.90	6.50/6/16	6.16	6.51
Think Times						
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.02	12.02	10.01	5.02	5.02	N/A
Maximum	120.23	120.22	100.11	50.23	50.21	N/A
Keying Times						
Minimum	18.00	3.00	2.00	2.00	2.00	N/A
Average	18.01	3.01	2.01	2.01	2.01	N/A
Maximum	18.06	3.06	2.05	2.05	2.05	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.

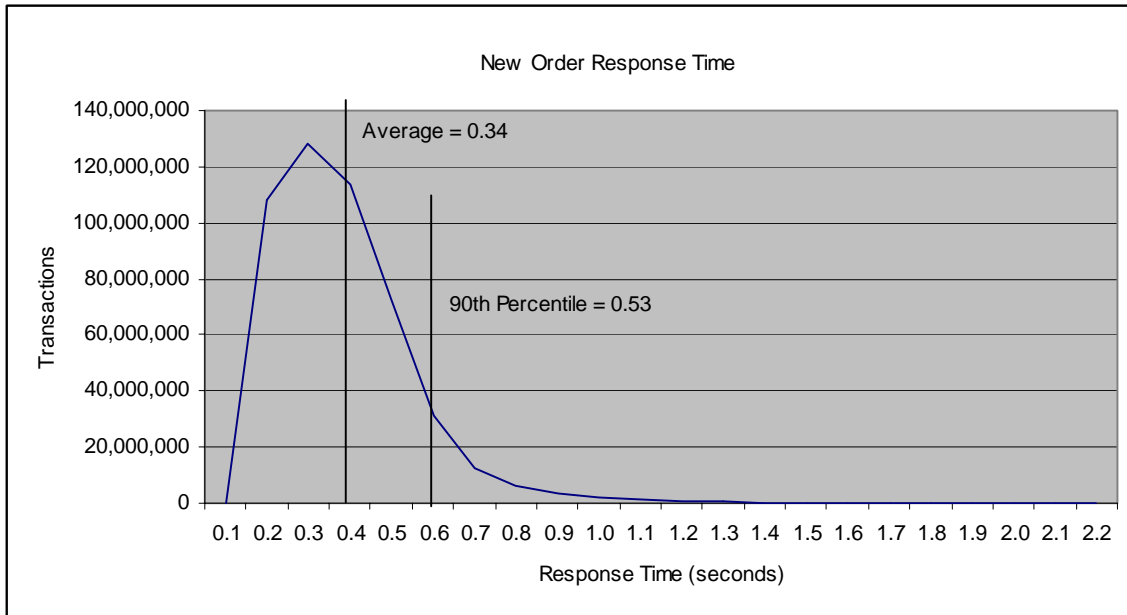


Figure 5-1: New-Order Response Time Distribution

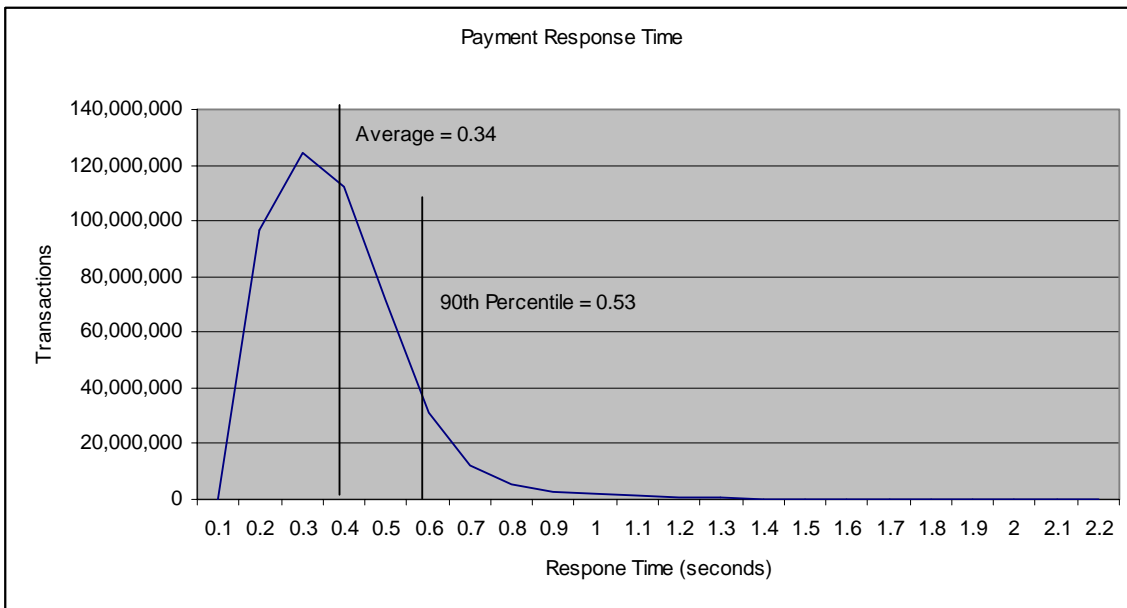


Figure 5-2: Payment Response Time Distribution

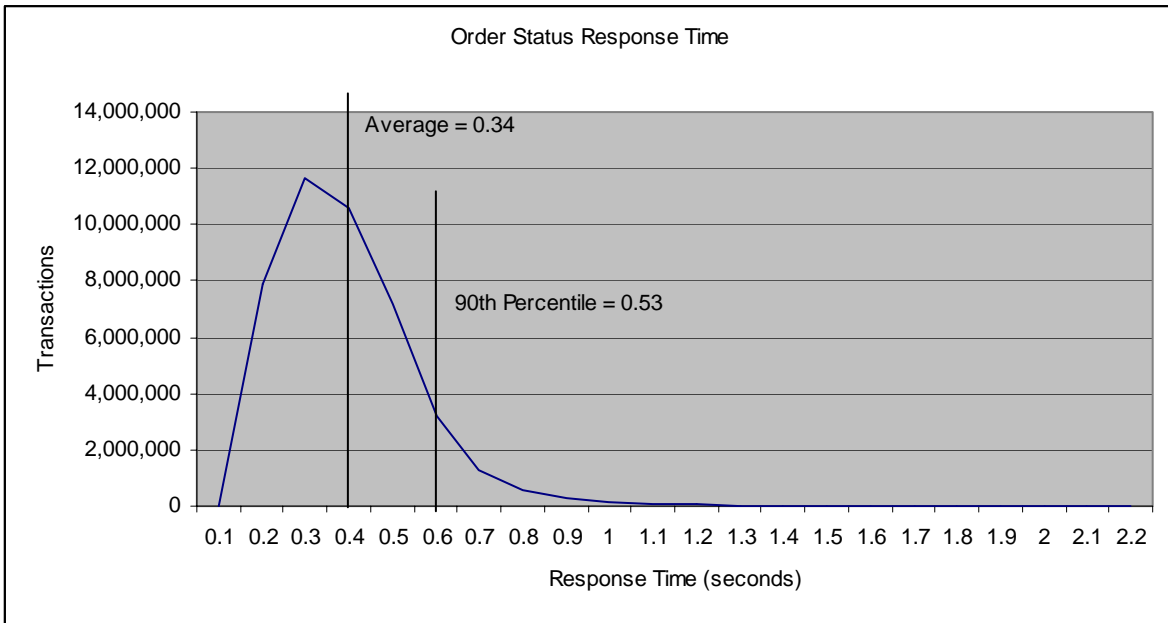


Figure 5-3: Order-Status Response Time Distribution

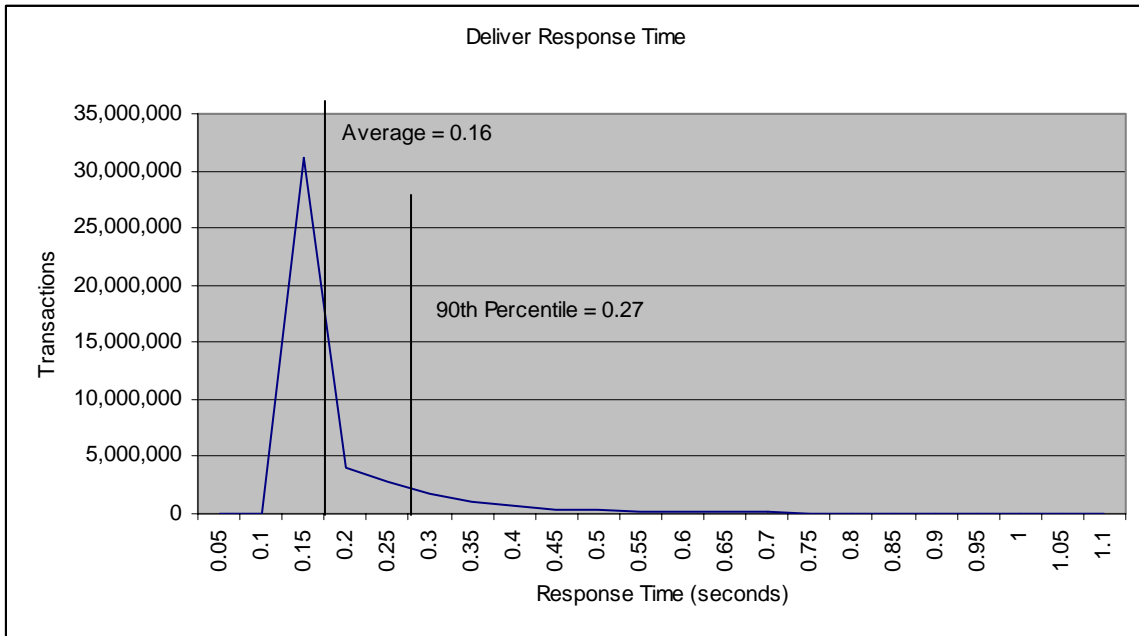


Figure 5-4: Delivery (Interactive) Response Time Distribution

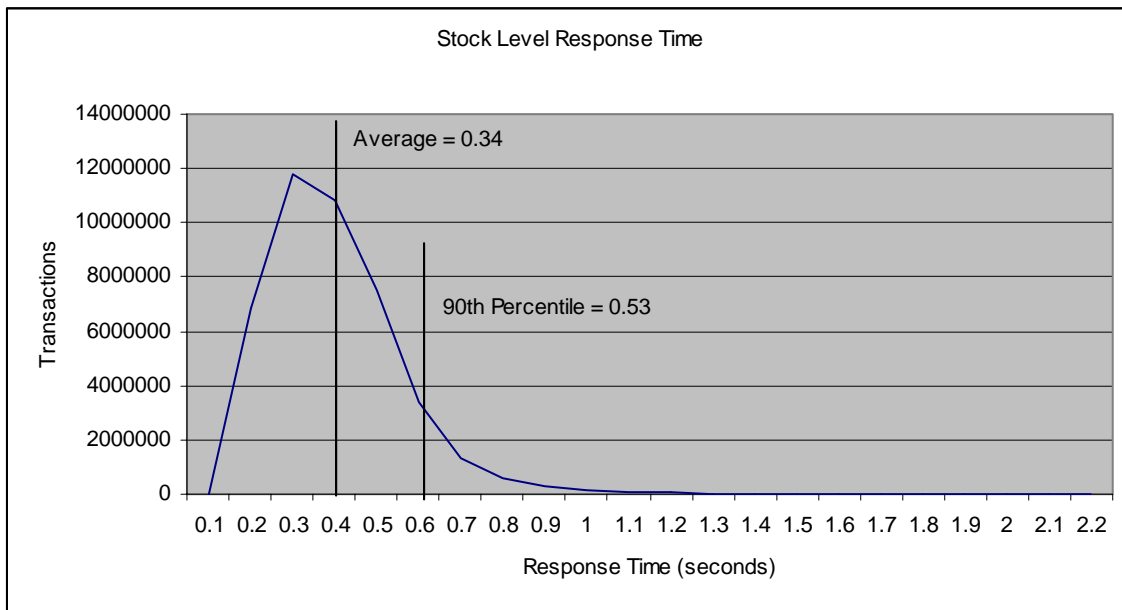


Figure 5-5: Stock Level Response Time Distribution

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.

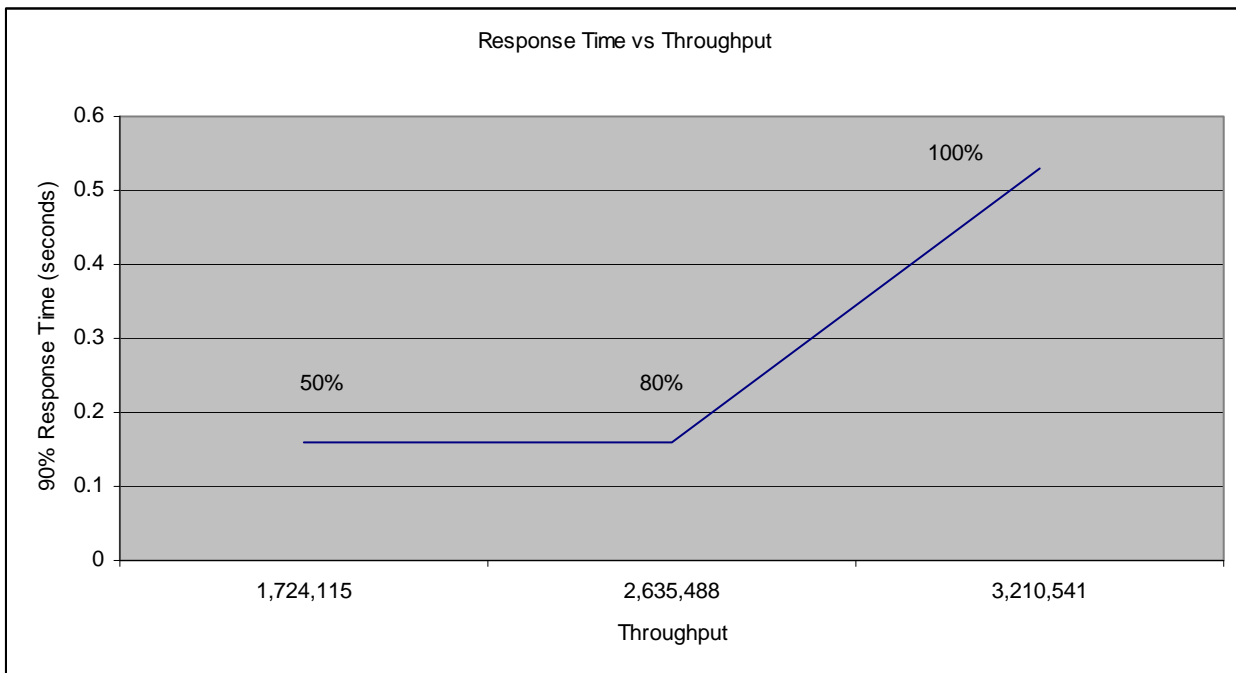


Figure 5-6: New-Order Response Time vs. Throughput

5.5. Think Time Frequency Distribution

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

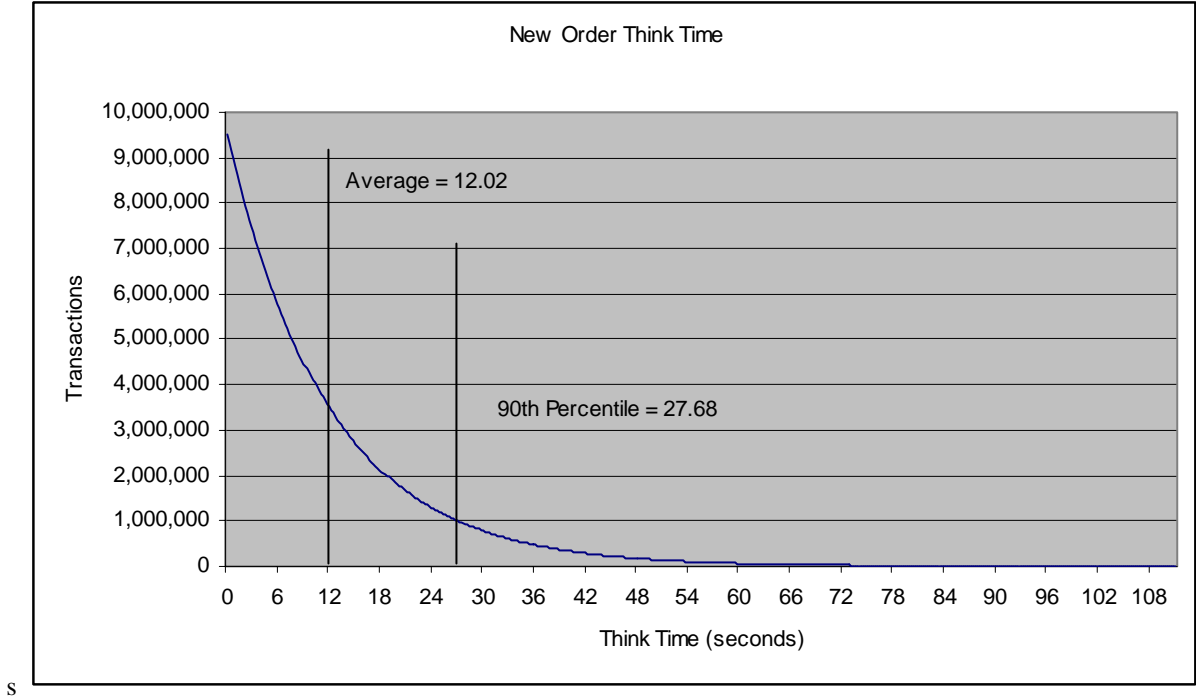


Figure 5-7: New-Order Think Time Distribution

5.6. Throughput versus Elapsed Time

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

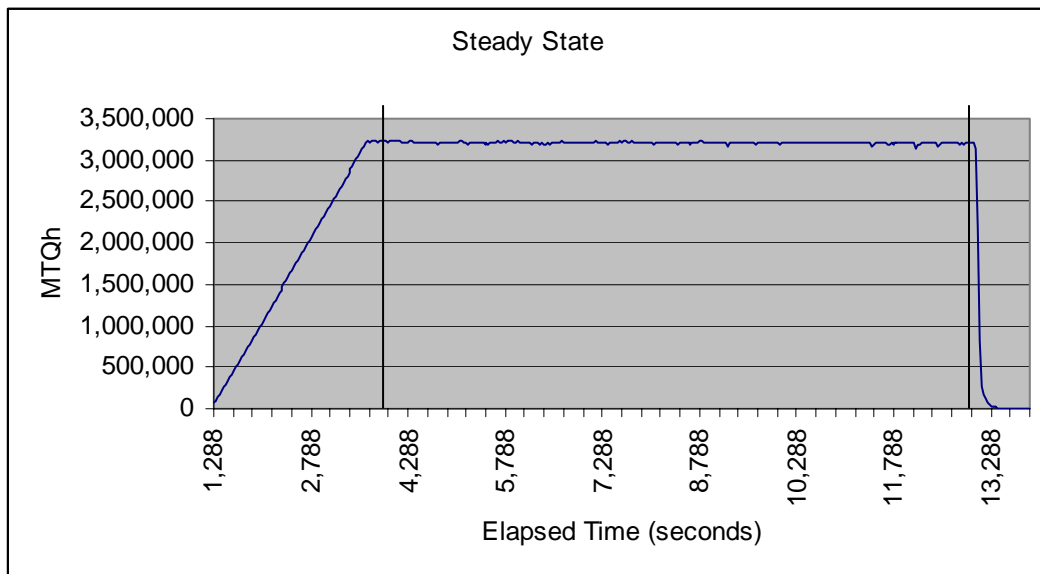


Figure 5-8: New-Order Throughput vs. Elapsed Time

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 30-minute measurement interval was used to guaranty 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 2000 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 the its unique warehouse and district to the IIS ISAPI handler. Upon successful validation of user's login, IIS the displays an HTML form which encapsulates the TPC-C transaction menu.

The transaction flow is described below:

- The TPC-C user requests the transaction type's HTML form and proceeds to generate (fill in) a GET request with the required files for the transaction.
- IIS accepts the filled in GET request , parses, and validates all values entered by the user.
- It then proceeds to transmit those values to the COM+ server through an transaction type specific COM+ api interface.
- The COM+ Pool Manager receives the request and first decides if there is a connection object in the pool available to service it.
 - If so, the connection is used to send the transaction request to the Server.
 - If no connection is available, the request will enter a COM+ internal queue and will be serviced by the next available connection.
- Once the connection is available to be used, a COM+ pool thread receives the transaction and calls a TPC-C back end DB2 client api to execute all database operations related to the transaction type. (All the transaction information entered on the HTML form is available in a data structure provided by the ISAPI caller).
- The transaction is committed and the DB2 back end client returns control back to the COM pool thread.
- COM pool thread returns control to the ISAPI caller.
(All transaction results are inside the data structure that the ISAPI caller provided to the COM+ api in the parameter list).
- ISAPI caller returns control to the "screen application" by doing a PUT request.

5.8.2. Database Transaction

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

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

DB2 UDB Server proceeds to update the database as follows:

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

5.8.3. Checkpoints

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

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

5.9. Measurement Interval

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

A 2-hour 30-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 8 Ethernet Gigabit switches each with a rate of 1000Mbits full-duplex. Each group of 16 clients were connected to one of the Gigabit Ethernet switches at 1000Mbits full-duplex rate.

6.4. Operator Intervention

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

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

7 Clause 7: Pricing Related Items

7.1. Hardware and Programs Used

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

The detailed list of all hardware and programs for the priced configuration is listed in the pricing sheets (please refer to Section 8.2 for details) for each system reported. The prices for all products and features that are provided by IBM are available the same day as product or feature availability.

7.2. Three Year Cost of System Configuration

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

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

Discount are based on US list prices and for similar quantities and configurations.

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 products are generally available today except the following:

Product	Availability Date
IBM eServer p5 595 Model 9119-595	April 8, 2005
DB2 Universal Database 8.2	May 14, 2005

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 eServer p5 595 Model 9119-595	3,210,540.63	\$16,273,320 USD	\$5.07 USD	May 14, 2005

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

8 Clause 9: Audit Related Items

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

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

Sponsor: John J. Makis
 IBM eServer Performance
 11501 Burnet Road
 Austin, TX 78758

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

November 15, 2004

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

Platform: IBM eServer p5 595 Model 9119-595 c/s
 Operating system: AIX 5L V5.3
 Database Manager: DB2 UDB 8.2
 Transaction Manager: Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	NewOrder Response Time - 90%	tpmC
Server: IBM eServer p5 595 Model 9119-595				
64 x POWER5 (1.9 GHz)	2048 GB main (32x 36 MB L3 cache)	8 x 36.4GB SCSI int. 6400 x 36.4GB FAStT 140 x 73.4GB FAStT	0.53 Sec.	3,210,540.63
128 Clients: IBM eServer xSeries 336 Model 883722U (Specification for each)				
2 x Intel Xeon (3.2 GHz)	2 GB main (1 MB L2 cache/cpu)	1 x 80GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 5.3 of the benchmark.

The following verification items were given special attention:

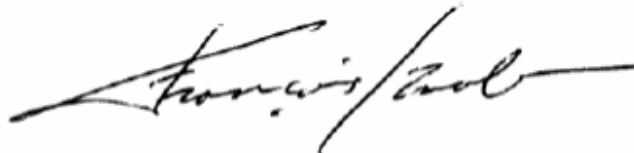
- The transactions were correctly implemented.
- The database records were the proper size.
- The database was properly scaled and populated.
- The ACID properties were met.

- Input data was generated according to the specified percentages.
- The transaction cycle times included the required browser delay, 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 2.5 hours.
- 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", with a long horizontal flourish extending to the right.

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 - 2004
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####
#
# Makefile.config - NT/Win2000 Makefile Configuration
#
# Make Configuration (MSVC)
MAKE=nmake.exe
# Compiler Configuration (MSVC).
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:\Program Files\Microsoft Visual Studio\VC98\Lib"
db2api.lib winmm.lib
LD_FLAGS_OUT=/OUT:
# Library Configuration
AR=lib.exe
ARFLAGS=
ARFLAGS_LIB=
ARFLAGS_OUT=/OUT:
# OS Commands
ERASE=del /F
ERASEDIR=rmdir /S
MOVE=MOVE
COPY=COPY
# OS File Extensions & Path Separator
```

```
OBJEXT=.obj
LIBEXT=.lib
SHLIBEXT=.dll
BINEXT=.exe
SLASH=\
CMDSEP=&
```

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 - 2004
@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=CK040318
@REM The DB2 Instance Name (for DB2)
set DB2INSTANCE=%USERNAME%
@REM The OS being used (i.e. "UNIX", "WINDOWS")
set PLATFORM=WINDOWS
@REM The type of make command and slash used by the OS
@REM (i.e. UNIX - "/", WINDOWS - "\")
@REM These are referenced all over the kit.
set SLASH=
set MAKE=nmake
@REM Specifies whether or not to use dari stored proc's for the
TPC-C driver. Set to either DARIVERSION or NONDARI;
@REM set TPCC_SPTYPE=NOSP
@REM set TPCC_SPTYPE=SPGENERAL2
set TPCC_SPTYPE=SPGENERAL
@REM set TPCC_SPTYPE=DARI2SQLDA
set DB2VERSION=v8
@REM The schema name is typically the SQL authorization ID
(or username).
@REM This is required for runstats and EEE.
set TPCC_SCHEMA=%USERNAME%
@REM DB2 EE/EEE Configuration
set DB2EDITION=EE
@REM set DB2EDITION=EEE
set DB2NODE=0
@REM set to the number of nodes you have. Set to 1 for EE.
set DB2NODES=1
@REM TPCC General Configuration
```

```
@REM ** IMPORTANT NOTE **
@REM The kit is not guaranteed to work properly if
TPCC_ROOT or TPCC_SQLLIB
@REM have spaces in them. If you absolutely must use paths
with spaces,
@REM then the entire path must be surrounded by double
quotes.
@REM For example: HOME="C:\Program Files\IBM"
set HOME=C:\home\tpcc
set TPCC_DBNAME=TPCC
set TPCC_ROOT=%HOME%\tpc-c.ibm
set TPCC_SQLLIB=C:\Progra~1\IBM\sqllib
set TPCC_RUNDATA=%HOME%\tpc-c.ibm\tpccdata
@REM TPCC Debug Configuration
@REM This is the path where all error and debug logs are
placed.
@REM To get debugging from within the stored procedures,
you must
@REM set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
set TPCC_DEBUGDIR=c:\temp
@REM Specifies where stored procedures should be placed
and if they should
@REM be fenced.
set TPCC_SPDIR=%TPCC_SQLLIB%\function
set TPCC_FENCED=NO
```

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 - 2004
** 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>
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];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    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 {
        int32_t s_I_PRICE;
        int32_t s_OL_AMOUNT;
        int16_t s_S_QUANTITY;
        int16_t pad2;
        char s_I_NAME[25];
        char s_brand_generic;
    } item[15];
    int64_t s_O_ENTRY_D_time;
    int32_t s_W_TAX;
    int32_t s_D_TAX;
    int32_t s_C_DISCOUNT;
    int32_t 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];
};
struct in_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time; /* init by SUT */
    int64_t 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];
    int64_t s_H_DATE_time;
    int64_t s_C_SINCE_time;
    int64_t s_C_CREDIT_LIM;
    int64_t s_C_BALANCE;
    int32_t s_C_DISCOUNT;
    int32_t s_C_ID;
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_W_STREET_1[21];
    char s_W_STREET_2[21];
    char s_W_CITY[21];
    char s_W_STATE[3];
    char s_W_ZIP[10];

```

```

    char s_D_STREET_1[21];
    char s_D_STREET_2[21];
    char s_D_CITY[21];
    char s_D_STATE[3];
    char s_D_ZIP[10];
    char s_C_FIRST[17];
    char s_C_MIDDLE[3];
    char s_C_LAST[17];
    char s_C_STREET_1[21];
    char s_C_STREET_2[21];
    char s_C_CITY[21];
    char s_C_STATE[3];
    char s_C_ZIP[10];
    char s_C_PHONE[17];
    char s_C_CREDIT[3];
    char s_C_DATA[201];
};
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];
    int64_t s_C_BALANCE;
    int64_t s_O_ENTRY_D_time;
    int32_t s_C_ID;
    int32_t s_O_ID;
    int16_t s_O_CARRIER_ID;
    int16_t s_ol_cnt;
    int16_t pad1[2];
    struct oitems_struct {
        int64_t s_OL_DELIVERY_D_time;
        int32_t s_OL_AMOUNT;
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad2;
    } item[15];
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_C_FIRST[17];
    char s_C_MIDDLE[3];
    char s_C_LAST[17];
};
struct in_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_O_DELIVERY_D_time; /* init by SUT */
    int32_t s_W_ID;
    int16_t s_O_CARRIER_ID;
};
struct out_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int32_t s_O_ID[10];
    int16_t s_transtatus;

```

```

int16_t deadlocks;
};
struct in_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_threshold;
int32_t s_W_ID;
int16_t s_D_ID;
};
struct out_stocklev_struct {
int16_t len;
int16_t pad[SPGENERAL_PAD];
int32_t s_low_stock;
int16_t s_transtatus;
int16_t deadlocks;
};
/*
*****
*/
/* Transaction Prototypes */
/*
*****
*/
#ifdef __cplusplus
extern "C" {
#endif
extern int neword_sql(struct in_neword_struct*, struct
out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct
out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct
out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct
out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct
out_stocklev_struct*);
#ifdef __cplusplus
}
#endif
/*
*****
*/
/* DB2 Connect/Disconnect & Thread Context Wrappers
*/
/*
*****
*/
#ifdef __cplusplus
extern "C" {
#endif
extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*, char*, char*);
extern int disconnect_from_TM(void);
extern int create_context(void);
extern int destroy_context(void);
extern int get_context(void**);
extern int attach_context(void*);
extern int detach_context(void*);
#ifdef __cplusplus
}
#endif
#endif // __DB2TPCC_H

```

include/lval.h

```

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 256000
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

```

include/tpccapp.h

```

/*
*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 -
2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM
Corp.
*****
*****
/*
* tpccapp.h - Application Macros
*/
#ifndef __TPCCAPP_H
#define __TPCCAPP_H
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "sqlenv.h"
#define daricall __stdcall
#include "sqlca.h"
#include "sqlcodes.h"
#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var) SwapEndian((void*)&Var, sizeof(Var))
/*
*****
*****
FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int I=0x12345678; SWAP_BYTE(I); I => 0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by
XOR op
e.g.: *a = 0x12 [ Addr + 0];
      *b = 0x78 [ Add + 4 - 0 - 1 = Addr+3];
      *a ^= *b; // sets *a to 0x6A
      *b ^= *a; // sets *b to 0x12
      *a ^= *b; // sets *a to 0x78
Now *a => 0x78 && *b => 0x12

```

```

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

/*
*****
*****
** SQLCODE Macros */
*****
#define DLCHK(a) \
if (sqlca.sqlcode == SQL_RC_E911) { goto a; }
/*
*****
*****
/* In NOT ATOMIC COMPOUND SQL, all statements will be
executed, but not
/*
/* all will 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:
HHHXNNNSSSSXNNNSSSS...
/*
/* (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 - 2004  
** 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 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
```

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 -  
2004  
## All Rights Reserved.  
##  
## US Government Users Restricted Rights - Use, duplication or  
## disclosure restricted by GSA ADP Schedule Contract with IBM  
Corp.  
*****  
*****  
#  
# Makefile - Makefile for Src.Common  
#  
!include $(TPCC_ROOT)/Makefile.config  
#  
*****  
*****  
# Preprocessor, Compiler and LInker Flags  
#  
*****  
*****  
BND_OPTS = GRANT PUBLIC \  
            MESSAGES $*.bnd.msg  
PRP_OPTS = BINDFILE \  
            OPTLEVEL 1 \  
            ISOLATION RR \  
            MESSAGES $*.prep.msg \  
            LEVEL $(TPCC_VERSION) \  
            NOLINEMACRO  
INCLUDES = -I$(TPCC_SQLLIB)$(SLASH)include -  
            I$(TPCC_ROOT)$(SLASH)include  
CFLAGS = $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDES) \  
          -DSQLA_NOLINES -D$(DB2EDITION) -  
          D$(DB2VERSION) \  
          -D$(TPCC_SPTYPE)  
UTIL_OBJ = tpccdbg$(OBJEXT) tpccctx$(OBJEXT)  
#  
*****  
*****  
# User Targets  
#  
*****  
*****  
all: connect $(UTIL_OBJ) disconnect  
clean:  
- $(ERASE) *$(OBJEXT) *.bnd *.msg tpccctx.c
```

```
#  
*****  
*****  
# Helper Targets  
#  
*****  
*****  
connect:  
- db2 connect to $(TPCC_DBNAME)  
disconnect:  
- db2 connect reset  
- db2 terminate  
  
rebind:  
db2 bind tpccctx.bnd $(BND_OPTS)  
#  
*****  
*****  
# Build Rules  
#  
*****  
*****  
.SUFFIXES:  
.SUFFIXES: $(OBJEXT) .c .sqc  
.sqc.c:  
@echo "Prepping $*.sqc"  
-db2 prep $*.sqc $(PRP_OPTS)  
@echo "Binding $*.bnd"  
db2 bind $*.bnd $(BND_OPTS)  
#  
*****  
*****  
# Dependencies  
#  
*****  
*****  
# Source  
tpccdbg$(OBJEXT): tpccdbg.c  
tpccctx$(OBJEXT): tpccctx.c  
tpccmisc$(OBJEXT): tpccmisc.c  
# Headers  
tpccdbg.c: $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Common/tpccctx.sqc

```
/******  
*****  
** Licensed Materials - Property of IBM  
**  
** Governed under the terms of the International  
** License Agreement for Non-Warranted Sample Code.  
**  
** (C) COPYRIGHT International Business Machines Corp.  
1996 - 2004  
** 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 <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>
#include "db2tpcc.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; }
    strncpy(dbname,in_dbname,8);
    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 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****
****/
/*
 * tccdbg.c - Debugging Routines
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"
#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128
void del_print();
void new_print();
void ord_print();
void pay_print();
void stk_print();
void current_tmstamp(char *buf);
static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";
/*-----*/
/* InitializeDebug */
/*-----*/
__inline void InitializeDebug(void) {
    if (debugInit == 0) {
        char *p = getenv("TPCC_DEBUGDIR");
        if (p) {
            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 */
sqlaintp(errStr, 512, 78, psqlca);
err_fp = fopen(err_fn, "a+");
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 (!isprint(c)) c = ' ';
            }
            fprintf(err_fp, "%c", c);
        }
        fprintf(err_fp, "\n");
        if (j < 4) fprintf(err_fp, " ");
    }
}
fprintf(err_fp, "sqlerrp: ");
for(j = 0; j < 8; j++)
    fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
fprintf(err_fp, "\n");
fprintf(err_fp, "sqlerrd: ");
for(j = 0; j < 6; j++)
    fprintf(err_fp, "%d", psqlca->sqlerrd[j]);
fprintf(err_fp, "\n");
if (psqlca->sqlwarn[0] != ' ')
{
    fprintf(err_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
        fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
    fprintf(err_fp, "\n");
}
fprintf(err_fp, "\n");
fclose(err_fp);
}
/*-----*/
/* del_debug          */
/*-----*/
/*-----*/
void del_debug (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct *in_delivery,
                char *msg)
{
    char debug_fn[DEBUG_PATH_SIZE +
DEBUG_FILENAME_SZ];

    InitializeDebug();
    strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);

```

```

    strcat(debug_fn, "del.debug.out");
    del_print(delivery_ptr, in_delivery, debug_fn, msg);
}
/*-----*/
/* del_print          */
/*-----*/
void del_print (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct *in_delivery,
                char *filename,
                char *msg)
{
    FILE *debug_fp;
    char timeStamp[27];
    int j;
    current_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=====");
    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, "  ts_O_DELIVERY_D = %lld (%lX)\n",
in_delivery->s_O_DELIVERY_D_time, in_delivery-
>s_O_DELIVERY_D_time);
    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);
}

```



```

fprintf(debug_fp, "\n=====\n");
fprintf(debug_fp, "in_stocklev_struct {\n");
fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
in_stocklev->s_W_ID, in_stocklev->s_W_ID);
fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
in_stocklev->s_D_ID, in_stocklev->s_D_ID);
fprintf(debug_fp, "ts_threshold = %d (%X)\n",
in_stocklev->s_threshold, in_stocklev->s_threshold);
fprintf(debug_fp, ")\n\n");
fprintf(debug_fp, "out_stocklev_struct {\n");
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
stocklev->s_transtatus, stocklev->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
stocklev->deadlocks, stocklev->deadlocks);
fprintf(debug_fp, "ts_low_stock = %d (%X)\n",
stocklev->s_low_stock, stocklev->s_low_stock);
fprintf(debug_fp, ")\n\n");
fclose(debug_fp);
}
void current_tmstamp(char *buf)
{
time_t t = time(NULL);
strncpy(buf, ctime(&t), 19);
}

```

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 - 2004
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####
# Makefile - Makefile for Src.Cli (RTE/Driver Interface)
#
!include $(TPCC_ROOT)/Makefile.config
#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####
BND_OPTS = GRANT PUBLIC \
MESSAGES $*.bnd.msg
PRP_OPTS = BINDFILE \
ISOLATION RR \

```

```

EXPLAIN ALL \
MESSAGES $*.prep.msg \
LEVEL $(TPCC_VERSION) \
NOLINEMACRO
INCLUDES = -I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include
CFLAGS = $(CFLAGS_OS) $(INCLUDES)
$(CFLAGS_DEBUG) \
$(UOPTS) -D$(DB2EDITION) -D$(DB2VERSION) -
D$(TPCC_SPTYPE)
OBJS =
$(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccctx$(OBJEXT) \
tpcccli$(OBJEXT)
LIBS = tpcccli$(LIBEXT)
#
#####
#####
# User Targets
#
#####
#####
all: connect $(OBJS) plan $(LIBS) disconnect
$(AR) $(ARFLAGS) $(ARFLAGS_OUT)tpcccli$(LIBEXT)
$(OBJS) $(ARFLAGS_LIB)
@echo "-----"
@echo "Please copy lval.h, db2tpcc.h, and
tpcccli$(LIBEXT) to"
@echo "a place where they can be #included and linked
with the"
@echo "RTE/driver code."
@echo "-----"
clean:
- $(ERASE) *.msg *.bnd *.plan *$(OBJEXT) *$(LIBEXT)
tpcccli.c
#
#####
#####
# Helper Targets
#
#####
#####
connect:
- db2 connect to $(TPCC_DBNAME)
disconnect:
- db2 connect reset
- db2 terminate
plan:
- db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -
s $(TPCC_SCHEMA) -w -1 -n TPCCCLI -g # 0 -o
TPCCCLI.exfmt.plan
- db2expln -d $(TPCC_DBNAME) -c $(TPCC_SCHEMA) -
p TPCCCLI -s 0 -g -o TPCCCLI.expln.plan
rebind: connect
db2 bind tpcccli.bnd $(BND_OPTS) QUERYOPT 7
#
#####
#####
# Build Rules

```

```

#
#####
#####
.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc
tpcccli.c:
@echo "Prepping $*.sqc"
-db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
@echo "Binding $*.bnd"
db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7
#
#####
#####
# Dependencies
#
#####
#####
# Client Library:
tpcccli$(LIBEXT): $(OBJS)
# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c
# Headers
tpcc_all_sql.c: $(TPCC_ROOT)/include/db2tpcc.h
$(TPCC_ROOT)/include/lval.h

```

Src.Cli/tpcccli.sqc

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2004
** 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"
#include "lval.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 diff item id then sort on that.
// If real/quasi dup, then sort on warehouse id.
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[ 270 ];
    } * pHostvarInput;
    struct vc_new_out
    {
        short len;
        char data[ 662 ];
    } * pHostvarOutput ;
    EXEC SQL END DECLARE SECTION;
    int clientRc = TRAN_OK ;
    int itemIndex = 0 ;
    int actualItemIndex = 0 ;
    /* Create Timestamp */
    in_neword->s_O_ENTRY_D_time = time(NULL) ;
    // Sort the item list ; This helps eliminate duplicates anyway,
    and since invalid item
    // IDs == 100001 , we will remain compliant with 2.4.2.3
    Comment 1.
    // !! So DON'T sort or change the order of the items in any
    subsequent code, including SQL.
    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 ;
        }
    }
    // Pre-duplicate processing count
    in_neword->s_O_OL_CNT = itemIndex ;
    // Sort the original array
    qsort( in_neword->in_item, in_neword->s_O_OL_CNT
, sizeof ( in_neword->in_item[ 0 ] )
, itemComparison
);
    // Now purge the duplicates.
    actualItemIndex = -1 ;

```

```

for ( itemIndex = 0
; itemIndex < in_neword->s_O_OL_CNT
; itemIndex++ )
{
    // If duplicate, just increment the item entry order count
    if ( actualItemIndex >= 0
&& in_neword->in_item[ actualItemIndex ].s_OL_I_ID
== in_neword->in_item[ itemIndex ].s_OL_I_ID
&& in_neword-
>in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID ==
in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID
)
    {
        in_neword->in_item[ actualItemIndex ].s_OL_QUANTITY
+= in_neword->in_item[ itemIndex ].s_OL_QUANTITY ;
    }
    else
    {
        actualItemIndex ++ ;
        in_neword->in_item[ actualItemIndex ].s_OL_I_ID =
in_neword->in_item[ itemIndex ].s_OL_I_ID ;
        in_neword-
>in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID = in_neword-
>in_item[ itemIndex ].s_OL_SUPPLY_W_ID ;
        in_neword->in_item[ actualItemIndex ].s_OL_QUANTITY
= in_neword->in_item[ itemIndex ].s_OL_QUANTITY ;
    }
}
in_neword->s_O_OL_CNT = actualItemIndex + 1 ;
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_O_ENTRY_D_time);
    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_O_ENTRY_D_time);
    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_O_ENTRY_D_time);
    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 )
{
    double wtax = neword->s_W_TAX / 10000.0 ;
    double dtax = neword->s_D_TAX / 10000.0 ;
    double cdisc = neword->s_C_DISCOUNT / 10000.0 ;
    double factor = (1.0 - cdisc) * (1.0 + wtax + dtax) ;
    // Post process the item set, detecting any bad items , and
    set or count from that.
    // Anything that could be deferred from the SP to the client
    has been.
    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 ;
        }
    }
    // s_total_amount gets cast implicitly to a double to do the
arithmetic,

```

```

// and then cast back to a sqlint32.
neword->s_total_amount *= factor;
}
else
{
    sqlerror( NEWORD_SQL, "NEW", __FILE__, __LINE__,
&sqlca );
neword->s_transtatus = FATAL_SQLERROR ;
clientRc = FATAL_SQLERROR ;
}
/* Update Output Structure with Timestamp */
neword->s_O_ENTRY_D_time = in_neword-
>s_O_ENTRY_D_time ;
#ifdef DEBUGIT
new_debug(neword, in_neword, "Client after SP call");
#endif /* DEBUGIT */
if (neword->s_transtatus <= FATAL_SQLERROR)
{
    new_debug(neword, in_neword, "NEW failed");
    clientRc = FATAL_SQLERROR ;
}
if (neword->s_transtatus == INVALID_ITEM)
{
    clientRc = INVALID_ITEM ;
}
return ( clientRc ) ;
}
// -----
// Payment CLIENT
// -----
int payment_sql ( struct in_payment_struct * in_payment
, struct out_payment_struct * payment )
{
    struct sqlca sqlca ;
    int clientRc = TRAN_OK ;
    EXEC SQL BEGIN DECLARE SECTION;
    // Inputs
    sqlint64 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 ;
    sqlint64 h_date ;
    // Outputs
    sqlint32 c_id ;
    sqlint64 c_credit_lim ;
    sqlint32 c_discount ;
    sqlint64 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 ] ;
    sqlint64 c_since ;
    char c_data [ 200 ] ;

```

```

short c_data_indicator = 0 ;
    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
#define h_date in_payment->s_H_DATE_time
// 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
/* Create Timestamp */
in_payment->s_H_DATE_time = (sqlint64) time( NULL ) ;
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 )
{
    // Pre-built c_data prefix for BC does not include c_id in this
instance
    // Tne strange $04.4d.%02.2d printf modifier is to print a
(4,2)
    // 0-padded floating-point value -- %f won't 0-pad by default.

```

```

c_data_prefix_c_last.len =
sprintf( c_data_prefix_c_last.data, "%2.2d %6.6d %2.2d %6.6d
%04.4d.%02.2d", c_d_id, c_w_id, d_id, w_id, (int)(h_amount
/ 100), (int)(h_amount % 100) );
// 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 ;

// Pre-built c_data prefix for BC does include c_id in this
instance
// Tne strange $04.4d.%02.2d printf modifier is to print a
(4,2)
// 0-padded floating-point value -- %f won't 0-pad by default.
c_data_prefix_c_id.len = sprintf( c_data_prefix_c_id.data, "
%5.5d %2.2d %6.6d %2.2d %6.6d %04.4d.%02.2d", c_id ,
c_d_id, c_w_id, d_id, w_id, (int)(h_amount / 100),
(int)(h_amount % 100) ) ;
}
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

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_li
m
, :c_discount, :c_balance, :c_data :c_data_indicator

FROM TABLE ( PAY_C_LAST( :w_id
, :d_id
, :c_w_id
, :c_d_id
, :c_last_input
, :h_date
, :h_amount
, :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
    )
    ;
    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

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_li
m
        , :c_discount , :c_balance , :c_data :c_data_indicator

FROM TABLE ( PAY_C_ID( :w_id
        , :d_id
        , :c_w_id
        , :c_d_id
        , :in_c_id
        , :h_date
        , :h_amount
        , :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
    )
    ;
    COMMIT ;
END COMPOUND ;
}
/* Update Output Structure with Timestamp */
payment->s_H_DATE_time = in_payment->s_H_DATE_time ;
#endif DEBUGIT

```

```

    pay_debug(payment, in_payment, "Client after SQL call");
#endif /* DEBUGIT */
    if ( sqlca.sqlcode != 0 )
    {
        DLCHK( retry_tran ) ;
        sqlerror( PAYMENT_SQL , "PAY" , __FILE__ , __LINE__ ,
&sqlca ) ;
        payment->s_transtatus = FATAL_SQLERROR ;
        clientRc = FATAL_SQLERROR ;
        pay_debug( payment, in_payment, "PAY failed" ) ;
        EXEC SQL ROLLBACK WORK ;
        if ( sqlca.sqlcode != 0 )
        {
            sqlerror( PAYMENT_SQL, "ROLLBACK FAILED",
__FILE__ , __LINE__ , &sqlca ) ;
        }
    }
    return ( clientRc ) ;
}
// -----
// Order Status CLIENT
// -----

int ordstat_sql ( struct in_ordstat_struct * in_ordstat
        , struct out_ordstat_struct * ordstat )
{
    struct sqlca sqlca ;
    EXEC SQL BEGIN DECLARE SECTION;
    struct vc_ord_in
    {
        short len ;
        char data[ 42 ] ;
    } * in_ord ;
    struct vc_ord_out
    {
        short len ;
        char data[ 446 ] ;
    } * 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_O_ENTRY_D_time);
    SWAP_BYTE(ordstat->s_C_ID);
    SWAP_BYTE(ordstat->s_O_ID);
    SWAP_BYTE(ordstat->s_O_CARRIER_ID);
    SWAP_BYTE(ordstat->s_O_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_DELIVERY_D_time);
        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[ 22 ] ;
} * 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 ;
/* Create Timestamp */
in_delivery->s_O_DELIVERY_D_time = (sqlint64)
time( NULL ) ;
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_O_DELIVERY_D_time);
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_O_DELIVERY_D_time);
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 w_id ;
###short d_id ;
sqlint32 threshold ;
// output

sqlint32 low_stock ;
EXEC SQL END DECLARE SECTION;
#define w_id in_stocklev->s_W_ID
#define d_id in_stocklev->s_D_ID
#define threshold in_stocklev->s_threshold
#define low_stock stocklev->s_low_stock
stocklev->deadlocks = -1 ;
stocklev->s_transtatus = TRAN_OK ;
#ifdef DEBUGIT
stk_debug(stocklev, in_stocklev, "Client before SQL call");
#endif /* DEBUGIT */
retry_tran:
stocklev->deadlocks ++ ;
/*
EXEC SQL
SELECT ITEMS_BELOW_THRESHOLD
INTO :low_stock
FROM TABLE( STOCK_LEVEL( :w_id, :d_id, :threshold ) )
AS T
WITH CS;
*/

EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC
SELECT COUNT( S_I_ID ) INTO :low_stock
FROM ( SELECT DISTINCT S_I_ID
FROM ORDER_LINE , STOCK , DISTRICT
WHERE D_W_ID = :w_id
AND D_ID = :d_id
AND OL_O_ID < d_next_o_id
AND OL_O_ID >= ( d_next_o_id - 20 )
AND OL_W_ID = D_W_ID
AND OL_D_ID = D_ID
AND S_I_ID = OL_I_ID
AND S_W_ID = OL_W_ID
AND S_QUANTITY < :threshold

) OLS

WITH CS
;

COMMIT ;

END COMPOUND ;
#ifdef DEBUGIT
stk_debug(stocklev, in_stocklev, "Client after SQL call");
#endif /* DEBUGIT */
if ( sqlca.sqlcode != 0 )
{

```

```

DLCHK( retry_tran ) ;
sqlerror( STOCKLEV_SQL, "STK", __FILE__, __LINE__,
&sqlca);
stocklev->s_transtatus = FATAL_SQLERROR ;
clientRc = FATAL_SQLERROR ;
stk_debug( stocklev, in_stocklev, "STK failed" ) ;
EXEC SQL ROLLBACK WORK ;
if ( sqlca.sqlcode != 0 )
{
sqlerror( STOCKLEV_SQL, "ROLLBACK FAILED",
__FILE__, __LINE__, &sqlca ) ;
}
}
return ( clientRc ) ;
}

```

NULLDB/NULLDB.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 NULLDB_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
// NULLDB_API functions as being imported from a DLL,
whereas this DLL sees symbols
// defined with this macro as being exported.
#ifdef NULLDB_EXPORTS
#define NULLDB_API __declspec(dllexport)
#else
#define NULLDB_API __declspec(dllimport)
#endif
extern NULLDB_API int dataSet;
extern "C" NULLDB_API int do_nord(struct nord_wrapper
*nord,void *ctx);
extern "C" NULLDB_API int do_pymt(struct paym_wrapper
*pymt,void *ctx);
extern "C" NULLDB_API int do_orcs(struct orcs_wrapper
*orcs,void *ctx);
extern "C" NULLDB_API int do_dlv(struct dlv_wrapper
*dlv,void *ctx);
extern "C" NULLDB_API int do_stok(struct stok_wrapper
*stok,void *ctx);
extern "C" NULLDB_API int connect_db(char *dbName,void
**ctx);
extern "C" NULLDB_API int disconnect_db(void *ctx);

```

NULLDB/NULLDB.cpp

```

// NULLDB.cpp : Defines the entry point for the DLL application.
//
#include "stdafx.h"
#include "NULLDB.h"
#include "..\tpccsapi\tpcc.h"
BOOL WINAPI DllMain( HANDLE hModule,
DWORD ul_reason_for_call,
LPVOID lpReserved
)
{

```



```

switch (ul_reason_for_call)
{
case DLL_PROCESS_ATTACH:
case DLL_THREAD_ATTACH:
case DLL_THREAD_DETACH:
case DLL_PROCESS_DETACH:
break;
}
return TRUE;
}
// This is an example of an exported variable
NULLDB_API int dataSet = 0;
extern "C" NULLDB_API int connect_db(char *dbName, void
**ctx)
{
return OK;
}
extern "C" NULLDB_API int disconnect_db(void *ctx)
{
return OK;
}
extern "C" NULLDB_API int do_nord(struct nord_wrapper
*nord, void *ctx)
{
nord->out_nord.s_transtatus = 0;
if (dataSet == 0)
{
strcpy(nord->out_nord.s_C_LAST, "NOYOLA");
strcpy(nord->out_nord.s_C_CREDIT, "GC");
nord->out_nord.s_W_TAX = 1694;
nord->out_nord.s_D_TAX = 967;
nord->out_nord.s_C_DISCOUNT = 1024;
nord->out_nord.s_O_ID = 3013;
nord->out_nord.s_O_OL_CNT = 4;
nord->out_nord.s_total_amount = 32345;
nord->out_nord.s_O_ENTRY_D_time = 1234567890;
strcpy(nord->out_nord.item[0].s_I_NAME, "98 Toyota Supra
Turbo");
nord->in_nord.in_item[0].s_OL_I_ID = 1;
nord->in_nord.in_item[0].s_OL_QUANTITY = 1;
nord->in_nord.in_item[0].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[0].s_I_PRICE = 42000;
nord->out_nord.item[0].s_OL_AMOUNT = 554000;
nord->out_nord.item[0].s_S_QUANTITY = 31;
nord->out_nord.item[0].s_brand_generic = 'G';
strcpy(nord->out_nord.item[1].s_I_NAME, "HKS Turbo
Timer");
nord->in_nord.in_item[1].s_OL_I_ID = 1;
nord->in_nord.in_item[1].s_OL_QUANTITY = 1;
nord->in_nord.in_item[1].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[1].s_I_PRICE = 4500;
nord->out_nord.item[1].s_OL_AMOUNT = 438100;
nord->out_nord.item[1].s_S_QUANTITY = 57;
nord->out_nord.item[1].s_brand_generic = 'G';
strcpy(nord->out_nord.item[2].s_I_NAME, "TRD GEN2
Exhaust");
nord->in_nord.in_item[2].s_OL_I_ID = 1;
nord->in_nord.in_item[2].s_OL_QUANTITY = 1;
nord->in_nord.in_item[2].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[2].s_I_PRICE = 6734;
nord->out_nord.item[2].s_OL_AMOUNT = 47173;
nord->out_nord.item[2].s_S_QUANTITY = 42;

```

```

nord->out_nord.item[2].s_brand_generic = 'G';
strcpy(nord->out_nord.item[3].s_I_NAME, "BLITZ DUAL-
SOLENOID");
nord->in_nord.in_item[3].s_OL_I_ID = 1;
nord->in_nord.in_item[3].s_OL_QUANTITY = 1;
nord->in_nord.in_item[3].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[3].s_I_PRICE = 35000;
nord->out_nord.item[3].s_OL_AMOUNT = 12096;
nord->out_nord.item[3].s_S_QUANTITY = 84;
nord->out_nord.item[3].s_brand_generic = 'G';
dataSet = 1;
}
else
{
strcpy(nord->out_nord.s_C_LAST, "SIMPSON");
strcpy(nord->out_nord.s_C_CREDIT, "GC");
nord->out_nord.s_W_TAX = 913;
nord->out_nord.s_D_TAX = 1519;
nord->out_nord.s_C_DISCOUNT = 958;
nord->out_nord.s_O_ID = 1410;
nord->out_nord.s_O_OL_CNT = 9;
nord->out_nord.s_total_amount = 12345;
nord->out_nord.s_O_ENTRY_D_time = 1234567890;
strcpy(nord->out_nord.item[0].s_I_NAME, "97 Toyota Supra
NA");
nord->in_nord.in_item[0].s_OL_I_ID = 1;
nord->in_nord.in_item[0].s_OL_QUANTITY = 1;
nord->in_nord.in_item[0].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[0].s_I_PRICE = 30000;
nord->out_nord.item[0].s_OL_AMOUNT = 769600;
nord->out_nord.item[0].s_S_QUANTITY = 97;
nord->out_nord.item[0].s_brand_generic = 'G';
strcpy(nord->out_nord.item[1].s_I_NAME, "98 Turbo Stereo");
nord->in_nord.in_item[1].s_OL_I_ID = 1;
nord->in_nord.in_item[1].s_OL_QUANTITY = 1;
nord->in_nord.in_item[1].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[1].s_I_PRICE = 10001;
nord->out_nord.item[1].s_OL_AMOUNT = 192999;
nord->out_nord.item[1].s_S_QUANTITY = 51;
nord->out_nord.item[1].s_brand_generic = 'G';
strcpy(nord->out_nord.item[2].s_I_NAME, "XERD Exhaust
Header");
nord->in_nord.in_item[2].s_OL_I_ID = 1;
nord->in_nord.in_item[2].s_OL_QUANTITY = 1;
nord->in_nord.in_item[2].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[2].s_I_PRICE = 4000;
nord->out_nord.item[2].s_OL_AMOUNT = 41670;
nord->out_nord.item[2].s_S_QUANTITY = 14;
nord->out_nord.item[2].s_brand_generic = 'G';
strcpy(nord->out_nord.item[3].s_I_NAME, "LEXOL
Conditioner");
nord->in_nord.in_item[3].s_OL_I_ID = 1;
nord->in_nord.in_item[3].s_OL_QUANTITY = 1;
nord->in_nord.in_item[3].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[3].s_I_PRICE = 1400;
nord->out_nord.item[3].s_OL_AMOUNT = 17213;
nord->out_nord.item[3].s_S_QUANTITY = 90;
nord->out_nord.item[3].s_brand_generic = 'G';
strcpy(nord->out_nord.item[4].s_I_NAME, "TRD Sticker 1");
nord->in_nord.in_item[4].s_OL_I_ID = 1;
nord->in_nord.in_item[4].s_OL_QUANTITY = 1;
nord->in_nord.in_item[4].s_OL_SUPPLY_W_ID = 1;

```

```

nord->out_nord.item[4].s_I_PRICE = 1400;
nord->out_nord.item[4].s_OL_AMOUNT = 27232;
nord->out_nord.item[4].s_S_QUANTITY = 75;
nord->out_nord.item[4].s_brand_generic = 'G';
strcpy(nord->out_nord.item[5].s_I_NAME, "TRD Sticker 2");
nord->in_nord.in_item[5].s_OL_I_ID = 1;
nord->in_nord.in_item[5].s_OL_QUANTITY = 1;
nord->in_nord.in_item[5].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[5].s_I_PRICE = 4400;
nord->out_nord.item[5].s_OL_AMOUNT = 35808;
nord->out_nord.item[5].s_S_QUANTITY = 22;
nord->out_nord.item[5].s_brand_generic = 'G';
strcpy(nord->out_nord.item[6].s_I_NAME, "TRD Sticker 3");
nord->in_nord.in_item[6].s_OL_I_ID = 1;
nord->in_nord.in_item[6].s_OL_QUANTITY = 1;
nord->in_nord.in_item[6].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[6].s_I_PRICE = 5500;
nord->out_nord.item[6].s_OL_AMOUNT = 44392;
nord->out_nord.item[6].s_S_QUANTITY = 21;
nord->out_nord.item[6].s_brand_generic = 'G';
strcpy(nord->out_nord.item[7].s_I_NAME, "TRD Sticker 4");
nord->in_nord.in_item[7].s_OL_I_ID = 1;
nord->in_nord.in_item[7].s_OL_QUANTITY = 1;
nord->in_nord.in_item[7].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[7].s_I_PRICE = 8300;
nord->out_nord.item[7].s_OL_AMOUNT = 83410;
nord->out_nord.item[7].s_S_QUANTITY = 35;
nord->out_nord.item[7].s_brand_generic = 'G';
strcpy(nord->out_nord.item[8].s_I_NAME, "98 Toyota OEM
Bra");
nord->in_nord.in_item[8].s_OL_I_ID = 1;
nord->in_nord.in_item[8].s_OL_QUANTITY = 1;
nord->in_nord.in_item[8].s_OL_SUPPLY_W_ID = 1;
nord->out_nord.item[8].s_I_PRICE = 10000;
nord->out_nord.item[8].s_OL_AMOUNT = 43160;
nord->out_nord.item[8].s_S_QUANTITY = 73;
nord->out_nord.item[8].s_brand_generic = 'G';
dataSet = 0;
}
return OK;
}
extern "C" NULLDB_API int do_pymt(struct paym_wrapper
*pymt, void *ctx)
{
pymt->out_paym.s_transtatus = 0;
if (dataSet == 0)
{
pymt->out_paym.s_C_CREDIT_LIM = 5000000;

pymt->out_paym.s_C_DISCOUNT = 1024;
pymt->out_paym.s_C_BALANCE = 17815;
pymt->out_paym.s_C_ID = 89;
pymt->out_paym.s_H_DATE_time = 1234567890;
strcpy(pymt->out_paym.s_W_STREET_1, "11501 Burnet
Rd");
strcpy(pymt->out_paym.s_W_STREET_2, "BLD 905");
strcpy(pymt->out_paym.s_W_CITY, "Austin");
strcpy(pymt->out_paym.s_W_STATE, "TX");
strcpy(pymt->out_paym.s_W_ZIP, "78758");
strcpy(pymt->out_paym.s_D_STREET_1, "11900 Hobby
Horse");
strcpy(pymt->out_paym.s_D_STREET_2, "Apt. 525");

```

```

strcpy(pyamt->out_paym.s_D_CITY,"Valley");
strcpy(pyamt->out_paym.s_D_STATE,"TX");
strcpy(pyamt->out_paym.s_D_ZIP,"78559");
strcpy(pyamt->out_paym.s_C_FIRST,"Jim");
strcpy(pyamt->out_paym.s_C_MIDDLE,"F");
strcpy(pyamt->out_paym.s_C_LAST,"Truck");
strcpy(pyamt->out_paym.s_C_STREET_1,"100 N Solis");
strcpy(pyamt->out_paym.s_C_STREET_2,"Flat 343");
strcpy(pyamt->out_paym.s_C_CITY,"Cambridge");
strcpy(pyamt->out_paym.s_C_STATE,"NY");
strcpy(pyamt->out_paym.s_C_ZIP,"785585432");
strcpy(pyamt->out_paym.s_C_PHONE,"1234567890123456");
pyamt->out_paym.s_C_SINCE_time = 0;
strcpy(pyamt->out_paym.s_C_CREDIT,"BC");
strcpy(pyamt-
>out_paym.s_C_DATA,"XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXX");
    dataSet = 1;
}
else
{
    pyamt->out_paym.s_C_CREDIT_LIM = 4000000;
    pyamt->out_paym.s_C_DISCOUNT = 52400;
    pyamt->out_paym.s_C_BALANCE = 14080;
    pyamt->out_paym.s_C_ID = 3180;
    pyamt->out_paym.s_H_DATE_time = 1234567890;
    strcpy(pyamt->out_paym.s_W_STREET_1,"1201 Park Ave.");
    strcpy(pyamt->out_paym.s_W_STREET_2,"Suite 432");
    strcpy(pyamt->out_paym.s_W_CITY,"Denver");
    strcpy(pyamt->out_paym.s_W_STATE,"CO");
    strcpy(pyamt->out_paym.s_W_ZIP,"787562356");
    strcpy(pyamt->out_paym.s_D_STREET_1,"3404 Garth Rd");
    strcpy(pyamt->out_paym.s_D_STREET_2,"Suite 320");
    strcpy(pyamt->out_paym.s_D_CITY,"Austin");
    strcpy(pyamt->out_paym.s_D_STATE,"TX");
    strcpy(pyamt->out_paym.s_D_ZIP,"785598767");
    strcpy(pyamt->out_paym.s_C_FIRST,"John");
    strcpy(pyamt->out_paym.s_C_MIDDLE,"P");
    strcpy(pyamt->out_paym.s_C_LAST,"Williams");
    strcpy(pyamt->out_paym.s_C_STREET_1,"North Rab Road");
    strcpy(pyamt->out_paym.s_C_STREET_2,"Apt 343");
    strcpy(pyamt->out_paym.s_C_CITY,"La Fiera");
    strcpy(pyamt->out_paym.s_C_STATE,"TX");
    strcpy(pyamt->out_paym.s_C_ZIP,"785585432");
    strcpy(pyamt->out_paym.s_C_PHONE,"1234567890123456");
    pyamt->out_paym.s_C_SINCE_time = 0;
    strcpy(pyamt->out_paym.s_C_CREDIT,"GC");
    strcpy(pyamt->out_paym.s_C_DATA,"Great Ebay");
    dataSet = 0;
}
return OK;
}
extern "C" NULLDB_API int do_orcls(struct orcls_wrapper
*orcls,void *ctx)
{
    orcls->out_orcls.s_transtatus = 0;
    if (dataSet == 0)
    {

```

```

orcls->out_orcls.s_C_BALANCE = 100000;
orcls->out_orcls.s_C_ID = 3;
orcls->out_orcls.s_O_ID = 1696;
orcls->out_orcls.s_O_CARRIER_ID = 9;
orcls->out_orcls.s_ol_cnt = 6;
orcls->out_orcls.s_O_ENTRY_D_time = 1234567890;
strcpy(orcls->out_orcls.s_C_FIRST,"Homer");
strcpy(orcls->out_orcls.s_C_MIDDLE,"J");
strcpy(orcls->out_orcls.s_C_LAST,"Simpson");
orcls->out_orcls.item[0].s_OL_AMOUNT = 30000;
orcls->out_orcls.item[0].s_OL_I_ID = 23492;
orcls->out_orcls.item[0].s_OL_SUPPLY_W_ID = 9;
orcls->out_orcls.item[0].s_OL_QUANTITY = 5;
orcls->out_orcls.item[0].s_OL_DELIVERY_D_time =
1234567890;
orcls->out_orcls.item[1].s_OL_AMOUNT = 12300;
orcls->out_orcls.item[1].s_OL_I_ID = 18860;
orcls->out_orcls.item[1].s_OL_SUPPLY_W_ID = 9;
orcls->out_orcls.item[1].s_OL_QUANTITY = 5;
orcls->out_orcls.item[1].s_OL_DELIVERY_D_time =
1234567890;
orcls->out_orcls.item[2].s_OL_AMOUNT = 15000;
orcls->out_orcls.item[2].s_OL_I_ID = 90488;
orcls->out_orcls.item[2].s_OL_SUPPLY_W_ID = 9;
orcls->out_orcls.item[2].s_OL_QUANTITY = 5;
orcls->out_orcls.item[2].s_OL_DELIVERY_D_time =
1234567890;
orcls->out_orcls.item[3].s_OL_AMOUNT = 25000;
orcls->out_orcls.item[3].s_OL_I_ID = 22741;
orcls->out_orcls.item[3].s_OL_SUPPLY_W_ID = 9;
orcls->out_orcls.item[3].s_OL_QUANTITY = 5;
orcls->out_orcls.item[3].s_OL_DELIVERY_D_time =
1234567890;
orcls->out_orcls.item[4].s_OL_AMOUNT = 20000;
orcls->out_orcls.item[4].s_OL_I_ID = 92952;
orcls->out_orcls.item[4].s_OL_SUPPLY_W_ID = 9;
orcls->out_orcls.item[4].s_OL_QUANTITY = 5;
orcls->out_orcls.item[4].s_OL_DELIVERY_D_time =
1234567890;
orcls->out_orcls.item[5].s_OL_AMOUNT = 2345;
orcls->out_orcls.item[5].s_OL_I_ID = 29956;
orcls->out_orcls.item[5].s_OL_SUPPLY_W_ID = 9;
orcls->out_orcls.item[5].s_OL_QUANTITY = 5;
orcls->out_orcls.item[5].s_OL_DELIVERY_D_time =
1234567890;
    dataSet = 1;
}
else
{
    orcls->out_orcls.s_C_BALANCE = 123000;
    orcls->out_orcls.s_C_ID = 856;
    orcls->out_orcls.s_O_ID = 418;
    orcls->out_orcls.s_O_CARRIER_ID = 10;
    orcls->out_orcls.s_ol_cnt = 5;
    strcpy(orcls->out_orcls.s_C_FIRST,"Erick");
    strcpy(orcls->out_orcls.s_C_MIDDLE,"J");
    strcpy(orcls->out_orcls.s_C_LAST,"Forman");
    orcls->out_orcls.s_O_ENTRY_D_time = 1234567890;
    orcls->out_orcls.item[0].s_OL_AMOUNT = 12000;
    orcls->out_orcls.item[0].s_OL_I_ID = 54602;
    orcls->out_orcls.item[0].s_OL_SUPPLY_W_ID = 10;
    orcls->out_orcls.item[0].s_OL_QUANTITY = 5;

```

```

    orcls->out_orcls.item[0].s_OL_DELIVERY_D_time =
1234567890;
    orcls->out_orcls.item[1].s_OL_AMOUNT = 2300;
    orcls->out_orcls.item[1].s_OL_I_ID = 18860;
    orcls->out_orcls.item[1].s_OL_SUPPLY_W_ID = 10;
    orcls->out_orcls.item[1].s_OL_QUANTITY = 5;
    orcls->out_orcls.item[1].s_OL_DELIVERY_D_time =
1234567890;
    orcls->out_orcls.item[2].s_OL_AMOUNT = 56009;
    orcls->out_orcls.item[2].s_OL_I_ID = 90488;
    orcls->out_orcls.item[2].s_OL_SUPPLY_W_ID = 10;
    orcls->out_orcls.item[2].s_OL_QUANTITY = 5;
    orcls->out_orcls.item[2].s_OL_DELIVERY_D_time =
1234567890;
    orcls->out_orcls.item[3].s_OL_AMOUNT = 98000;
    orcls->out_orcls.item[3].s_OL_I_ID = 22741;
    orcls->out_orcls.item[3].s_OL_SUPPLY_W_ID = 10;
    orcls->out_orcls.item[3].s_OL_QUANTITY = 5;
    orcls->out_orcls.item[3].s_OL_DELIVERY_D_time =
1234567890;
    orcls->out_orcls.item[4].s_OL_AMOUNT = 25000;
    orcls->out_orcls.item[4].s_OL_I_ID = 92952;
    orcls->out_orcls.item[4].s_OL_SUPPLY_W_ID = 10;
    orcls->out_orcls.item[4].s_OL_QUANTITY = 5;
    orcls->out_orcls.item[4].s_OL_DELIVERY_D_time =
1234567890;
    dataSet = 0;
}
return OK;
}
extern "C" NULLDB_API int do_dlv(struct dlv_wrapper
*dlv,void *ctx)
{
    dlv->out_dlv.s_transtatus = 0;
    if (dataSet == 0)
    {
        dataSet = 1;
        for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            dlv->out_dlv.s_O_ID[districtIndex]= 2055;
    }
    else
    {
        for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
            dlv->out_dlv.s_O_ID[districtIndex]= 2056;
        dataSet = 0;
    }
    return OK;
}
extern "C" NULLDB_API int do_stok(struct stok_wrapper
*stok,void *ctx)
{
    stok->out_stok.s_transtatus = 0;
    if (dataSet == 0)
    {
        stok->out_stok.s_low_stock = 100;
        dataSet = 1;
    }
    else
    {

```

```

    stok->out_stok.s_low_stock = 40;
    dataSet = 0;
}
return OK;
}

```

nullDB/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

```

nullDB/stdafx.cpp

```

// stdafx.cpp : source file that includes just the standard includes
// nullDB.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.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
{
    // Constructors / Destructor
public:
    htmlPhraser(char *queryString);
    ~htmlPhraser() {return;}

    // getters
public:
    int getCommandId();
    int validate(int txnType);

    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/resource.h

```

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

// Next default values for new objects
//
#ifdef 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/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
#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
#ifndef _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
transaxtions

```

```

//
// standard includes
#ifndef _COMMON_TPCC
#define _COMMON_TPCC
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/timeb.h>
#include <time.h>
#include <db2tpcc.h>
#include <iostream>
#include <fstream>
#include <process.h>
#include <ios>
// Defines
// OK
#define OK 0
// INVALID_STATUS
#define INVALID_STATUS -1
// ERR
#define ERR -1
// INVALID_COM_STATUS
#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
#define TXN_PAYMENT 2
#define TXN_ORDER_STATUS 3
#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:\r\n\r\nConnection: Keep-Alive\r\n\r\n"
// 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_DATE_TIME_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
#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
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 sqlint16;
typedef INT32b sqlint32;
typedef INT64b sqlint64;
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
#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
#define ERR_INVALID_TXN_TYPE

```

```

#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_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_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
// Registry Definitions
#define REGISTRY_SUB_KEY "SOFTWARE\\TPCC"
#define DELIVERY_THREADS "dlvyThreads"
#define DELIVERY_QUEUE_LEN "dlvyQueueLen"
#define DELIVERY_LOG_PATH "dlvyLogPath"
#define ERROR_LOG_FILE "errorLogFile"
#define HTML_TRACE_LOG_FILE "htmlTraceLogFile"
#define DB_NAME "dbName"
#define NULL_DB "nullDB"
#define COM_NULL_DB "comnullDB"
#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\\dlvy"
#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
// Structure defines
struct nord_wrapper {
    struct in_neword_struct in_nord;
    struct 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_ord;
    struct out_ordstat_struct out_ord;
};
struct dlvy_wrapper {
    struct in_delivery_struct in_dlvy;
    struct out_delivery_struct out_dlvy;
};
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 {
        double s_I_PRICE;
        double 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;
double 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;
};

// 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,INT64 *number);
inline int copyInMoney(const char *value);
inline void copyOutMoney64(char *buffer,INT64b
value,unsigned int len);
inline int copyOutDateTime(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++;
byteCount++;
}

//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)++ = ' ';
        }
    }
    **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 EPOC seconds
    int dateValue = ((int) (value & 0xfffffff)) +
(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 remaioning 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;
}
//
// 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 buff[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;
                }

```

```

        buff[j++] = value[i];
    }
}
int amount = atoi(buff);
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
    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;
    *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
calcOutDateTime(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++ = (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
    int debugFlag = 1;
#else
    int debugFlag = 0;
#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;
}
#define DEBUGADDRESS(POINTER) hex << (void *)
POINTER << dec
#define ERRORMSG(TEXT)
    EnterCriticalSection(&errorMutex);
    \
    debugFileName(__FILE__)
    \
    errorStream <<

```

```

    << "|" << __TIMESTAMP__
    << "| " << __LINE__ << "|" \
    << _getpid() << "|" <<
GetCurrentThreadId() << "|" \
    << TEXT;
    errorStream.flush();

    LeaveCriticalSection(&errorMutex);
#ifdef _DEBUG
    #define DEBUGMSG(TEXT)
    EnterCriticalSection(&debugMutex);

    debugStream <<
    debugFileName(__FILE__)
    \
    << "|" << __TIMESTAMP__
    << "| " << __LINE__ << "|" \
    << _getpid() << "|" <<
GetCurrentThreadId() << "|" \
    << TEXT;
    debugStream.flush();

    LeaveCriticalSection(&debugMutex);
#define DEBUGSTRING(TEXT,LENGTH)
    debugVarString(TEXT,LENGTH)

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

#endif
#endif /* _COMMON_TPCC */

```

tpccIsapi/tpccIsapi.def

```

; tpccIsapi.def : declares the module parameters for the DLL.
LIBRARY "tpccIsapi"
EXPORTS
    HttpExtensionProc
    GetExtensionVersion
    TerminateExtension

```

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 <httpext.h>
#include <tpcc.h>
#include <htmlPhraser.h>
#include <iomanip>
#include <db2tpcc.h>
#include <comsvcs.h>
// Terminal struct
// Terminal struct
// Terminal struct
struct TERM_ENTRY
{
    int terminalID;
    bool terminalInUse;
    int w_id;
    short d_id;
};
// COM interface
// COM interface
struct COM_HANDLE
{
    Itpcc_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;
    //user data
    int w_id;
    int d_id;
    int sync_id;
    int term_id;
    int conn_id;
    COM_HANDLE comInterface;
};
struct DLVYQUEUEDATA
{
    int warehouse;
    short in_s_0_CARRIER_ID;
    struct _timeb enqueueTime;
};
// Definitions
// Definitions
#define INVALID_ITEM 100
#define HEADER "Content-
Type:text/html\r\nContent-Length: %d\r\nConnection: Keep-
Alive\r\n\r\n"

```

```

#define      TLS_NULL          0xFFFFFFFF
#define      ACCESS_TIMEOUT    3600000
#define      DELIVERY_LOG_SUCCESS_STR  "--"
Tran %d Queue %d.%03d Start %d.%03dnW_ID: %d
CARRIER_ID: %d %s\nend-time: %d.%03dn"
// 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, short hour in milli seconds
int appendHiddenFields(char *htmlPage, TXN_HANDLE
*txnHandle);
int displayStatus(char *htmlPage, int rc);
#endif

```

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;
}
}

```



```

    {
    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/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/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>
// For custom assert and trace handling with WebDbg.exe
[ module(name="tpccIsapi", type="dll") ];
[ emitidl(restricted) ];
#define _WIN32_DCOM
/////////////////////////////////////////////////////////////////
// Globals
/////////////////////////////////////////////////////////////////
int      maxDataSize;           //max struct size
of all txn(s)
int      numUsers;             //number of users
that client will service.
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 DLVYQUEUEDATA *dlvyQueue;
//dlvy queue
HANDLE      *dlvyThreadHandles;
//ptr to array of thread handles
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 *pymt,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;

/////////////////////////////////////////////////////////////////
// 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];
            function failed.\n");
            sprintf(response,"ERROR: Init txnHandle
            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-
>lpszQueryString;

        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->lpszExtensionDesc, "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));
    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);
    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);

```

```

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);
}
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);

    return true;
}
extern "C" BOOL WINAPI TerminateExtension(DWORD
dwFlags)
{
    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);
    EnterCriticalSection(&isapiLock);

    HRESULT hres = NULL;
    try
    {
        DEBUGMSG("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);
        (*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);
        DEBUGMSG("Calling CoInitialize with txnHandle:
        "<<DEBUGADDRESS(*txnHandle)<<endl);
        hres =
        CoInitializeEx(NULL, COINIT_MULTITHREADED);
        if (FAILED(hres))
        {
            ERRORMSG("CoInitializeEx() failed, rc :
            "<<hres<<endl);
            return(ERR);
        };

        struct _timeb      startTime;
        struct _timeb      endTime;

        DEBUGMSG("Calling CoCreateInstance with
        txnHandle:"<<DEBUGADDRESS(*txnHandle)<< endl);
        _ftime(&startTime);
        hres =
        CoCreateInstance(CLSID_tpcc_com, NULL, CLSCTX_SERVER,
        ID_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);

    //call set complete to return object to pool.
    (*txnHandle)->comInterface.comHandle-
    >doSetComplete();
    //set the com buffers size
    DEBUGMSG("Setting txnHandle: " <<
    DEBUGADDRESS(*txnHandle) << "com buffer size to " <<
    maxDataSize<< endl)
    (*txnHandle)->comInterface.size = maxDataSize;
    DEBUGMSG("txnHandle:
    "<<DEBUGADDRESS(*txnHandle) <<"set to " << maxDataSize
    << endl);
    TlsSetValue(threadLSIndex, *txnHandle);
    DEBUGMSG("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
** Comments   :      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) )
    {
        return ERR_ORACLE_DLL_NOT_LOADED;
    }
    else
    {
        return ERR_UNKNOWN_DB;
    }
    db_connect =
    (CONNECT_PTR)GetProcAddress(dbInstance, "connect_db");
    if (db_connect == NULL)
    {
        return ERR_CONNECT_ADDRESS_NOT_FOUND;
    }
    dlvyCall =
    (DLVY_FUNC_PTR)GetProcAddress(dbInstance, "do_dlvY");
    if (dlvyCall == NULL)
    {
        return ERR_DLVY_ADDRESS_NOT_FOUND;
    }
}

```

```

return OK;
}
/*
*****
** Name      :      initDlvy
** Description :      initialize dlvy threads/dlvy queueu
** Parameters :
** Returns   :      int - return code
** Comments  :
*****
*/
int initDlvy()
{
    // Initialize critical section
    InitializeCriticalSection(&dlvyQueueLock);
    //create dlvy queue
    dlvyQueue = (DLVYQUEUEUEDATA *)
    calloc(dlvyQueueLen,sizeof(DLVYQUEUEUEDATA));
    dlvyThreadDone = CreateEvent(NULL,
                                TRUE,
                                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 txn queue
    dlvyBufferSlotIndex = 0;

    //thread index into dlvy txn queue
    dlvyBufferThreadIndex = 0;
    dlvyThreadHandles = new HANDLE[dlvyThreads];
    //create threads
    for(int threadCount = 0;threadCount <
dlvyThreads;threadCount++)
    {

```

```

        dlvyThreadHandles[threadCount] =
        (HANDLE)_beginthread(dlvyThreadEntry,0,NULL);
        if(dlvyThreadHandles[threadCount] ==
        INVALID_HANDLE_VALUE)
            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;

    //open up registry key
    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTER
Y_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_L
EN,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,&regT
ype,(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,&regT
ype,(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,&re
gType,(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,&regTy
pe,(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);
    RegCloseKey(registryKey);

return OK;
}
/*
*****
** Name      : doLoginForm
** Description : HTML Login page entry point
** Parameters :
**             htmlPhraser* command block
**             TXN_HANDLE* txn handle struct
** Returns    :
**             int - return code
** Comments   :
*****
*/
int doLoginForm(htmlPhraser *commandBlock, TXN_HANDLE
*txnHandle)
{
    DEBUGMSG("Entering doLoginForm()",<<endl);
    char *html=txnHandle->htmlPage;
    DEBUGMSG("Creating html login page",<<endl);
    //begin html page
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Client Home Page</TITLE></HEAD>"
        "<FORM ACTION=\\""
        APP_NAME
        "\" METHOD=\\"GET\\"">"
        "<H2>Please Login.</H2>"
        "<INPUT TYPE=\\"hidden\\""
NAME=\\""
        CMD_TXN_ID
        "\" VALUE=\\""
        CMD_MENU
        "\">"
        "<H3>Warehouse <INPUT
NAME=\\""
        CMD_W_ID
        "\" SIZE=6>"
        " District <INPUT NAME=\\""
        CMD_D_ID
        "\" SIZE=2></H3>"
        "<INPUT TYPE=\\"submit\\""
VALUE=\\"Submit\\"">"
        "</FORM>");

html+=sprintf(html,"divy Queue Length:%d <BR> num divy
threads:%d <BR> divy queue free slots:%d <BR> isapi queue
index:%d <BR> thread queue index:%d <BR>
</BODY></HTML>\n",
    divyQueueLen,
    divyThreads,
    divyBufferFreeSlots,
    divyBufferSlotIndex,
    divyBufferThreadIndex);
DEBUGMSG("Html login page done",<<endl);

```

```

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

    //validate parameters
    if( (txnHandle->w_id = atoi(commandBlock->get_W_ID()))
== 0 )
    {
        doLoginErrorPage(html,ERR_INVALID_W_ID);
        return OK;
    }
    if( (txnHandle->d_id = atoi(commandBlock->get_D_ID()))
== 0 )
    {
        doLoginErrorPage(html,ERR_INVALID_D_ID);
        return OK;
    }
    //store user into terminal array,
    //function will ERR if the terminal array is full
    if( assignTerminal(txnHandle) != OK)
    {
        doLoginErrorPage(html,ERR_TERMINAL_FULL);
        return OK;
    };
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Main Menu</TITLE></HEAD>\r\n"
        "<BODY><FORM ACTION=\\""
        APP_NAME
        "\" METHOD=\\"GET\\"">\r\n"
        "<H3>Please Select
Transaction.</H3>\r\n");
    html+=appendButtons(html);
    html+=appendHiddenFields(html,txnHandle);
    appendText(&html,"</FORM></BODY></HTML>");
    return OK;
}
/*
*****
** Name      : doLoginErrorPage
** Description : HTML Login page entry point
** Parameters :
**             char *
                 html page buffer
**             char *
                 error message
** Returns    :

```

```

**             int - return code
** Comments   :
*****
*/
int doLoginErrorPage(char *htmlPage,char *errorMessage)
{
    char *html=htmlPage;
    //begin html page
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Client Home Page</TITLE></HEAD>"
        "<FORM ACTION=\\""
        APP_NAME
        "\" METHOD=\\"GET\\"">"
        appendText(&html,"<H2>Please Login.</H2>"
        "<INPUT TYPE=\\"hidden\\""
NAME=\\""
        CMD_TXN_ID
        "\" VALUE=\\""
        CMD_MENU
        "\">"
        "<H3>Warehouse <INPUT
NAME=\\""
        CMD_W_ID
        "\" SIZE=6>"
        " District <INPUT NAME=\\""
        CMD_D_ID
        "\" SIZE=2></H3>"
        "<INPUT TYPE=\\"submit\\""
VALUE=\\"Submit\\"">"
        "</FORM>");
    appendText(&html,errorMessage);
    appendText(&html,"<BODY></HTML>");
    return OK;
}
/*
*****
** Name      : doNewOrderForm
** Description : HTML neworder page entry point
** Parameters :
**             htmlPhraser* command block
**             TXN_HANDLE* txn handle struct
** Returns    :
**             int - return code
** Comments   :
*****
*/
int doNewOrderForm(htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=txnHandle->htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD>\r\n"
        "<BODY><FORM ACTION=\\""
        APP_NAME
        "\" METHOD=\\"GET\\"">\r\n"
        "<CENTER><H3>Please Fill In
New Order Form.</H3></CENTER>\r\n" //check if not needed

```

```

        "Submit Transaction <INPUT
TYPE="submit" NAME="
        CMD_TXN_ID
        "\ VALUE="
        CMD_NORD
        "\>";
//append the hidden
html+=appendHiddenFields(html,txnHandle);

//int buffer for warehouse
char buffer[15];
appendText(&html," <PRE>"
//
//
//
5 6 7 8 9\r\n"
//
"123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890\r\n"
        "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>"
//
//
5 6 7 8 9\r\n"
//
"123456789012345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890\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);
//seal up html page
appendText(&html,"</PRE></BODY></HTML>");
return OK;
}
/*
*****
** Name      : doNewOrderResults
** Description :
** Parameters : HTML neworder page entry point
** Returns   :
** Comments  :
*****
*/
int doNewOrderResults(htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    DEBUGMSG("Entered doNewOrderResults" << endl);

```

```

char *html=txnHandle->htmlPage;
struct nord_wrapper *nord = NULL;
DEBUGMSG("Casting COM txnBuffer to nord struct"
<<endl);
nord = (nord_wrapper*)txnHandle-
>comInterface.txnBuffer;
ZeroMemory(nord,maxDataSize);
DEBUGMSG("COM txnBuffer initialized, validating input
parameters" << endl);

//set warehouse,district and customer id from command
block
nord->in_nord.s_W_ID = txnHandle->w_id;
DEBUGMSG("nord w_id:" << nord->in_nord.s_W_ID <<
endl);
if (nord->in_nord.s_D_ID = atoi(commandBlock-
>get_D_ID()) == 0)
{
    doNewOrderErrorPage(html,ERR_INVALID_D_ID,comma
ndBlock,txnHandle);
    return OK;
}
DEBUGMSG("nord d_id:" << nord->in_nord.s_D_ID <<
endl);
if((nord->in_nord.s_C_ID = atoi(commandBlock-
>get_C_ID())) == 0)
{
    doNewOrderErrorPage(html,ERR_INVALID_C_ID,comma
ndBlock,txnHandle);
    return OK;
}
DEBUGMSG("nord c_id:" << nord->in_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)) )
        if ( (nord->in_nord.in_item[nord-
>in_nord.s_O_OL_CNT].s_OL_SUPPLY_W_ID =
atoi(commandBlock->get_ITEM_SUPP_W(itemIndex))) == 0)
        {
            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)

```

```

        if ( (nord->in_nord.in_item[nord-
>in_nord.s_O_OL_CNT].s_OL_I_ID = atoi(commandBlock-
>get_ITEM_ITEM_NUM(itemIndex))) == 0)
        {
            doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,c
ommandBlock,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,c
ommandBlock,txnHandle);
            return OK;
        }
        //item qty
        if(*(commandBlock->get_ITEM_QTY(itemIndex))
        {
            if(itemComplete==2)
            {
                if( (nord->in_nord.in_item[nord-
>in_nord.s_O_OL_CNT].s_OL_QUANTITY =
atoi(commandBlock->get_ITEM_QTY(itemIndex))) == 0)
                {
                    doNewOrderErrorPage(html,ERR_INVALID_ITEM_OTY,c
ommandBlock,txnHandle);
                    return OK;
                }
                else
                    itemComplete++;
            }
            //missing previous value of item number
            else if( (itemComplete ==1)
            {
                doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,c
ommandBlock,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,c
ommandBlock,txnHandle);
    return OK;
}

DEBUGMSG("nord item:" << nord-
>in_nord.s_O_OL_CNT << "SUPPLY_W_ID:" << nord-
>in_nord.in_item[nord-
>in_nord.s_O_OL_CNT].s_OL_SUPPLY_W_ID <<
" OL_I_ID:" << nord->in_nord.in_item[nord-
>in_nord.s_O_OL_CNT].s_OL_I_ID << " OL_QUANTITY:" <<
nord->in_nord.in_item[nord-
>in_nord.s_O_OL_CNT].s_OL_QUANTITY <<endl);
if(itemComplete == 3)
nord->in_nord.s_O_OL_CNT++;
itemComplete=0;
}
DEBUGMSG("complete nord items:"<<nord-
>in_nord.s_O_OL_CNT<<" initializing remaing unused items "
<< NORD_ITEMS - nord->in_nord.s_O_OL_CNT << " to 0"
<<endl);
for(int itemIndex=nord-
>in_nord.s_O_OL_CNT;itemIndex<NORD_ITEMS;itemIndex++)
{
nord-
>in_nord.in_item[itemIndex].s_OL_SUPPLY_W_ID=0;
nord->in_nord.in_item[itemIndex].s_OL_I_ID = 0;
nord->in_nord.in_item[itemIndex].s_OL_QUANTITY
=0;
}
DEBUGMSG("nord creating new order results html title
page" <<endl);
appendText(&html,"<HTML><HEAD><TITLE>TPC-C New
Order Results</TITLE></HEAD>\r\n"
"<BODY><FORM ACTION=\""
APP_NAME
"\" METHOD=\"GET\">\r\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\r\n"
//
"12345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890\r\n
"";
//assume failure
nord->out_nord.s_transtatus = -1;

```

```

DEBUGMSG("nord executing COM interface function" <<
endl);
HRESULT hres;
try
{
hres = txnHandle->comInterface.comHandle-
>doNewOrder(&txnHandle-
>comInterface.size,(UCHAR**)&txnHandle-
>comInterface.txnBuffer);
}
catch(...)
{
html+=sprintf(html,"ERROR: nord com call caused
exeception to occur.</PRE></BODY></HTML>");
ERRORMSG("ERROR : nord com call cause
exeception to occur."<<endl);
return OK;
}
if(FAILED(hres))
{
ERRORMSG("ERROR : nord com call failed, rc:" <<
hex << hres);
DEBUGMSG("ERROR : nord com call failed, rc:" <<
hex << hres);
return OK;
}

//com call successful, return object back to pool.
hres = txnHandle->comInterface.comHandle-
>doSetComplete();
if(FAILED(hres))
{
ERRORMSG("ERROR : nord setcomplete call
failed, rc:" << hex << hres);
DEBUGMSG("ERROR : nord setcomplete call
failed, rc:" << hex << hres);
}
nord = (nord_wrapper *)txnHandle-
>comInterface.txnBuffer;
if(FAILED(hres))
{
html+=sprintf(html,"ERROR: nord com
doSetComplete failed, rc:%ld</PRE></BODY></HTML>",hres);
ERRORMSG("ERROR : nord com doSetComplete
failed, rc:"<<DEBUGADDRESS(hres)<<endl);
return OK;
}
DEBUGMSG("nord COM interface function successful,
s_transtatus:" << nord->out_nord.s_transtatus << endl);
int rc = nord->out_nord.s_transtatus;
char buffer[10];
appendText(&html,"Warehouse: ");
appendText(&html,itoa(nord-
>in_nord.s_W_ID,buffer,10),6,1);
appendText(&html,"District: ");
appendText(&html,itoa(nord-
>in_nord.s_D_ID,buffer,10),26,1);
appendText(&html,"Date: ");
if(rc == OK)
{
char dateTimeBuffer[50];

```

```

copyOutDateTime(dateTimeBuffer,nord-
>out_nord.s_O_ENTRY_D_time);
appendText(&html,dateTimeBuffer);
}
appendText(&html," <BR>"
"Customer: ");
appendText(&html,itoa(nord-
>in_nord.s_C_ID,buffer,10),8,1);
appendText(&html,"Name: ");
appendText(&html,nord-
>out_nord.s_C_LAST,LAST_NAME_LEN+3,1);
appendText(&html,"Credit: ");
appendText(&html,nord->out_nord.s_C_CREDIT,5,1);

appendText(&html,"%Disc.: ");
if(rc == OK)
{
html+=sprintf(html,"%2.2lf",nord-
>out_nord.s_C_DISCOUNT/100.0);
}
appendText(&html," <BR>"
"Order Number: ");
if(rc != INVALID_STATUS)
appendText(&html,itoa(nord-
>out_nord.s_O_ID,buffer,10),10,1);

appendText(&html,"Number of Lines: ");

if(rc != INVALID_STATUS)
appendText(&html,itoa(nord-
>out_nord.s_O_OL_CNT,buffer,10),10,1);
appendText(&html,"W_Tax: ");
if(rc == OK)
{
html+=sprintf(html,"%5.2lf",nord-
>out_nord.s_W_TAX/100.0);
}
appendText(&html," D_Tax: ");
if(rc == OK)
{
html+=sprintf(html,"%5.2lf",nord-
>out_nord.s_D_TAX/100.0);
}
appendText(&html," <BR> <BR>"
" 1 2 3 4 5 6
7 8 9\r\n"
//
"12345678901234567890123456789012345678901234567890
1234567890123456789012345678901234567890\r\n"
" Supp_W Item_Id Item_Name
Qty Stock B/G Price Amount <BR> ");

//display items
if (rc == OK)
{
//display valid items
for(int itemCount=0;itemCount < nord-
>out_nord.s_O_OL_CNT;itemCount++)
{

```



```

                "Credit Limit:<BR> <BR>Cust-
Data:<BR> <BR> <BR> <BR> </PRE>");
        return OK;
    }
}
/*
*****
** Name      : doPaymentResults
** Description :
** Parameters : HTML neworder page entry point
** Returns   :
** Comments  :
**           int - return code
**           :
**           :
*****
*/
int doPaymentResults(htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
    char *html=txnHandle->htmlPage;
    char buffer[50];
    struct paym_wrapper *pymt = NULL;
    pymt = (paym_wrapper*)txnHandle-
>comInterface.txnBuffer;
    ZeroMemory(pymt,maxDataSize);

    //set login warehouse id from command block
    pymt->in_paym.s_W_ID = txnHandle->w_id;
    //set district from command block
    if ( (pymt->in_paym.s_D_ID = atoi(commandBlock-
>get_D_ID())) == 0)
    {
        doPaymentErrorPage(html,ERR_INVALID_D_ID,comman
dBlock,txnHandle);
        return OK;
    }

    //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_CL
AST,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_ONL
Y,commandBlock,txnHandle);
        return OK;
    }
}
//get customer warehose id field
if ( (pymt->in_paym.s_C_W_ID = atoi(commandBlock-
>get_C_W_ID())) == 0)
{
    doPaymentErrorPage(html,ERR_INVALID_C_W_ID,com
mandBlock,txnHandle);
    return OK;
}
//get customer district id field
if ( (pymt->in_paym.s_C_D_ID = atoi(commandBlock-
>get_C_D_ID())) == 0)
{
    doPaymentErrorPage(html,ERR_INVALID_C_D_ID,comm
andBlock,txnHandle);
    return OK;
}
if (!copyInMoney64(commandBlock-
>get_AMT_PAID(),&pymt->in_paym.s_H_AMOUNT))
{
    doPaymentErrorPage(html,ERR_INVALID_PAYMENT_A
MOUNT,commandBlock,txnHandle);
    return OK;
}
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("Calling com entry api payment,
w_id:"<pymt->in_paym.s_W_ID<<" d_id:"<pymt-
>in_paym.s_D_ID<<endl);
    //assume failure
    pymt->out_paym.s_transtatus = -1;
    HRESULT hres;
    try
    {
        hres = txnHandle->comInterface.comHandle-
>doPayment(&txnHandle-
>comInterface.size,(UCHAR*)&txnHandle-
>comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: Com Payment call
caused exeception to occur.</PRE></BODY></HTML>");
        ERRORMSG("ERROR : Com Payment call caused
exeception to occur."<<endl);
        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);
        return OK;
    }
    hres = txnHandle->comInterface.comHandle-
>doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: pymt com
doSetComplete failed, rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : pymt com doSetComplete
failed, rc:"<<DEBUGADDRESS(hres)<<endl);
        return OK;
    }
    pymt = (paym_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>");
        ERRORMSG("Payment TXN ERROR"<<endl
<<"pymt->in_paym.s_C_D_ID:"<<pymt-
>in_paym.s_C_D_ID<<endl
<<"pymt->in_paym.s_C_ID:"<<pymt-
>in_paym.s_C_ID<<endl
<<"pymt->in_paym.s_C_LAST:"<<pymt-
>in_paym.s_C_LAST<<endl
<<"pymt->in_paym.s_C_W_ID:"<<pymt-
>in_paym.s_C_W_ID<<endl
<<"pymt->in_paym.s_D_ID:"<<pymt-
>in_paym.s_D_ID<<endl
<<"pymt->in_paym.s_H_AMOUNT:"<<pymt-
>in_paym.s_H_AMOUNT<<endl
<<"pymt->in_paym.s_H_DATE_time:"<<pymt-
>in_paym.s_H_DATE_time<<endl
<<"pymt->in_paym.s_W_ID:"<<pymt-
>in_paym.s_W_ID<<endl
<<"pymt->out_paym.deadlocks:"<<pymt-
>out_paym.deadlocks<<endl
<<"pymt->out_paym.s_C_BALANCE:"<<pymt-
>out_paym.s_C_BALANCE<<endl
<<"pymt->out_paym.s_C_CITY:"<<pymt-
>out_paym.s_C_CITY<<endl
<<"pymt->out_paym.s_C_CREDIT:"<<pymt-
>out_paym.s_C_CREDIT<<endl
<<"pymt->out_paym.s_C_CREDIT_LIM:"<<pymt-
>out_paym.s_C_CREDIT_LIM<<endl
<<"pymt->out_paym.s_C_DATA:"<<pymt-
>out_paym.s_C_DATA<<endl
<<"pymt->out_paym.s_C_DISCOUNT:"<<pymt-
>out_paym.s_C_DISCOUNT<<endl
<<"pymt->out_paym.s_C_FIRST:"<<pymt-
>out_paym.s_C_FIRST<<endl
<<"pymt->out_paym.s_C_ID:"<<pymt-
>out_paym.s_C_ID<<endl

```

```

        <<"pymt->out_paym.s_C_LAST:"<<pymt-
>out_paym.s_C_LAST<<endl
        <<"pymt->out_paym.s_C_MIDDLE:"<<pymt-
>out_paym.s_C_MIDDLE<<endl
        <<"pymt->out_paym.s_C_PHONE:"<<pymt-
>out_paym.s_C_PHONE<<endl
        <<"pymt->out_paym.s_C_SINCE_time:"<<pymt-
>out_paym.s_C_SINCE_time<<endl
        <<"pymt->out_paym.s_C_STATE:"<<pymt-
>out_paym.s_C_STATE<<endl
        <<"pymt->out_paym.s_C_STREET_1:"<<pymt-
>out_paym.s_C_STREET_1<<endl
        <<"pymt->out_paym.s_C_STREET_2:"<<pymt-
>out_paym.s_C_STREET_2<<endl
        <<"pymt->out_paym.s_C_ZIP:"<<pymt-
>out_paym.s_C_ZIP<<endl
        <<"pymt->out_paym.s_D_CITY:"<<pymt-
>out_paym.s_D_CITY<<endl
        <<"pymt->out_paym.s_D_STATE:"<<pymt-
>out_paym.s_D_STATE<<endl
        <<"pymt->out_paym.s_D_STREET_1:"<<pymt-
>out_paym.s_D_STREET_1<<endl
        <<"pymt->out_paym.s_D_STREET_2:"<<pymt-
>out_paym.s_D_STREET_2<<endl
        <<"pymt->out_paym.s_D_ZIP:"<<pymt-
>out_paym.s_D_ZIP<<endl
        <<"pymt->out_paym.s_H_DATE_time:"<<pymt-
>out_paym.s_H_DATE_time<<endl
        <<"pymt->out_paym.s_transtatus:"<<pymt-
>out_paym.s_transtatus<<endl
        <<"pymt->out_paym.s_W_CITY:"<<pymt-
>out_paym.s_W_CITY<<endl
        <<"pymt->out_paym.s_W_STATE:"<<pymt-
>out_paym.s_W_STATE<<endl
        <<"pymt->out_paym.s_W_STREET_1:"<<pymt-
>out_paym.s_W_STREET_1<<endl
        <<"pymt->out_paym.s_W_STREET_2:"<<pymt-
>out_paym.s_W_STREET_2<<endl
        <<"pymt->out_paym.s_W_ZIP:"<<pymt-
>out_paym.s_W_ZIP<<endl);
        return OK;
    }
    appendText(&html, "<BR><PRE>\n\n");
    // appendText(&html, " 1 2 3 4 5
6 7 8<BR>");
    //
    appendText(&html, "123456789012345678901234567890
12345678901234567890123456789012345678901234567890<
BR>");
    //start creating result body
    appendText(&html, "<BR><PRE>\n\n"
        "Date: ");
    copyOutDateTime(buffer,pymt-
>out_paym.s_H_DATE_time);
    appendText(&html,buffer);
    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: ");
        copyOutDateTime(buffer,pymt-
>out_paym.s_C_SINCE_time);
        appendText(&html,buffer);
        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,"%2.2lf",pymt-
>out_paym.s_C_DISCOUNT/100.0);

```

```

        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,"%-9.2lf",pymt-
>in_paym.s_H_AMOUNT/100.0);

        appendText(&html,"New Cust-Balance: $");
        html+=sprintf(html,"%-9.2lf",pymt-
>out_paym.s_C_BALANCE/100.0);

        appendText(&html,"<BR>Credit Limit: $");
        html+=sprintf(html,"%-9.2lf",pymt-
>out_paym.s_C_CREDIT_LIM/100.0);

        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>");
        return OK;
    }
    /*
    *****
    ** Name          : doPaymentErrorPage
    ** Description   :
    ** Parameters    : append payment error body
    **               : char *          html page result
    **               : char *          error message
    **               : htmlPhraser *  command block
    **               : TXN_HANDLE *   txn handle struct
    ** 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\">\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> <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 :
** HTML orderStatus page entry point
** Parameters :
** htmlPhraser* command block
** TXN_HANDLE* txn handle struct
** Returns :
** int - return code
** Comments :
**
*****
*/
int doOrderStatusForm(htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
char *html=txnHandle->htmlPage;
appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Order Status</TITLE></HEAD>\r\n"
"<BODY><FORM ACTION=\""
APP_NAME
\" METHOD=\"GET\">\r\n"
"<CENTER><H3>Please Fill In
Order Status Form.</H3></CENTER> <BR>\r\n"
"Submit Transaction <INPUT
TYPE=\"submit\" NAME=\""
CMD_TXN_ID
\" VALUE=\""
CMD_ORDS
\">");
html+=appendHiddenFields(html,txnHandle);
appendText(&html,"<PRE>\r\n"
"Warehouse: ");
char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));
appendText(&html,"
District: <INPUT NAME=\""
CMD_D_ID
\" SIZE=1>\r\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 :
** HTML orderStatus page entry point
** Parameters :

```

```

** htmlPhraser* command block
** char * html result page
** Returns :
** int - return code
** Comments :
**
*****
*/
int doOrderStatusResults(htmlPhraser
*commandBlock, TXN_HANDLE *txnHandle)
{
char *html=txnHandle->htmlPage;
struct ords_wrapper *ords = NULL;
ords = (ords_wrapper *) txnHandle-
>comInterface.txnBuffer;
ZeroMemory(ords,maxDataSize);
//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,com
mandBlock,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
_CLAST,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_CLAST_O
NLY,commandBlock,txnHandle);
return OK;
}
}
appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Order Status Results</TITLE></HEAD>\r\n"
"<BODY><FORM ACTION=\""
APP_NAME
\" METHOD=\"GET\">\r\n");
html+=appendButtons(html);
html+=appendHiddenFields(html,txnHandle);

```

```

appendText(&html,"</FORM>");
ords->out_ords.s_transtatus = -1;
HRESULT hres;
try
{
    hres = txnHandle->comInterface.comHandle-
doOrderStatus(&txnHandle-
comInterface.size,(UCHAR*)&txnHandle-
comInterface.txnBuffer);
}
catch(...)
{
    html+=sprintf(html,"ERROR: ords com call caused
exeception.</PRE></BODY></HTML>");
    return OK;
}
if(FAILED(hres))
{
    html+=sprintf(html,"ERROR: ords com call failed,
rc:%x</PRE></BODY></HTML>",hres);
    ERRORMSG("ERROR : ords com call failed,
rc:"<<DEBUGADDRESS(hres));
    return OK;
}
hres = txnHandle->comInterface.comHandle-
doSetComplete();
if(FAILED(hres))
{
    html+=sprintf(html,"ERROR: ords com
doSetComplete failed, rc:%ld</PRE></BODY></HTML>",hres);
    ERRORMSG("ERROR : ords com doSetComplete
failed, rc:"<<DEBUGADDRESS(hres)<<endl);
    return OK;
}
ords = (ords_wrapper *)txnHandle-
comInterface.txnBuffer;
int rc = ords->out_ords.s_transtatus;
if (rc != OK)
{
    html+=displayStatus(html,rc);
    appendText(&html,"</PRE></BODY></HTML>");
    ERRORMSG("ERROR order status"<<endl
<<"ords->in_ords.s_C_ID:"<<ords-
>in_ords.s_C_ID<<endl
<<"ords->in_ords.s_C_LAST:"<<ords-
>in_ords.s_C_LAST<<endl
<<"ords->in_ords.s_D_ID:"<<ords-
>in_ords.s_D_ID<<endl
<<"ords->in_ords.s_W_ID:"<<ords-
>in_ords.s_W_ID<<endl
<<"ords->out_ords.deadlocks:"<<ords-
>out_ords.deadlocks<<endl
<<"ords->out_ords.s_C_BALANCE:"<<ords-
>out_ords.s_C_BALANCE<<endl
<<"ords->out_ords.s_C_FIRST:"<<ords-
>out_ords.s_C_FIRST<<endl
<<"ords->out_ords.s_C_ID:"<<ords-
>out_ords.s_C_ID<<endl
<<"ords->out_ords.s_C_ID:"<<ords-
>out_ords.s_C_ID<<endl
<<"ords->out_ords.s_C_MIDDLE:"<<ords-
>out_ords.s_C_MIDDLE<<endl

```

```

<<"ords->out_ords.s_O_CARRIER_ID:"<<ords-
>out_ords.s_O_CARRIER_ID<<endl
<<"ords->out_ords.s_O_ENTRY_D_time:"<<ords-
>out_ords.s_O_ENTRY_D_time<<endl
<<"ords->out_ords.s_O_ID:"<<ords-
>out_ords.s_O_ID<<endl
<<"ords->out_ords.s_ol_cnt:"<<ords-
>out_ords.s_ol_cnt<<endl);
    return OK;
}
//start creating result body
appendText(&html,"</FORM><CENTER><H3>Order-
Status</H3></CENTER>");
appendText(&html, "<BR><PRE>\n\nWarehouse: ");
char buffer[50];

appendText(&html,itoa(ords-
>in_ords.s_W_ID,buffer,10),6+1,1);
appendText(&html,"District: ");
appendText(&html,itoa(ords->in_ords.s_D_ID,buffer,10));
appendText(&html,"<BR>"
"Customer: ");

//get customer id
appendText(&html,itoa(ords-
>in_ords.s_C_ID,buffer,10),6+1,1);
appendText(&html,"Name: ");
//get first, middle, and last from wrapper
appendText(&html,ords-
>out_ords.s_C_FIRST,FIRST_NAME_LEN+1,1);
appendText(&html,ords-
>out_ords.s_C_MIDDLE,INITIALS_LEN+1,1);
appendText(&html,ords-
>out_ords.s_C_LAST,LAST_NAME_LEN+5,1);
//get customer balance from wrapper
appendText(&html,"&#10;Cust-Balance: $");
html+=sprintf(html,"%2lf",ords-
>out_ords.s_C_BALANCE/100.0);
//display order number, entry date, and carrier number
appendText(&html,"<BR> <BR>"
"Order-Number ");
appendText(&html,itoa(ords-
>out_ords.s_O_ID,buffer,10),12,1);
appendText(&html,"Entry-Date: ");
copyOutDateTime(buffer,ords-
>out_ords.s_O_ENTRY_D_time);
appendText(&html,buffer,22,1);
appendText(&html,"Carrier-Number: ");
appendText(&html,itoa(ords-
>out_ords.s_O_CARRIER_ID,buffer,10));
//add item title columns
appendText(&html,"<BR>"
"Supply-W "
"Item-Id "
"Qty "
"Amount "
"Delivery-Date<BR> ");

//display items
for (int itemCount=0;itemCount<ords-
>out_ords.s_ol_cnt;itemCount++)
{
//appendSpaces(&html,2);

```

```

//get supp w
appendText(&html,itoa(ords-
>out_ords.item[itemCount].s_OL_SUPPLY_W_ID,buffer,10),11,
1);
//get item num
appendText(&html,itoa(ords-
>out_ords.item[itemCount].s_OL_I_ID,buffer,10),11,1);
//get item qty
appendText(&html,itoa(ords-
>out_ords.item[itemCount].s_OL_QUANTITY,buffer,10),6,1);
//get item dollar amount
html+=sprintf(html,"%14.2lf",ords-
>out_ords.item[itemCount].s_OL_AMOUNT/100.0);
//get delivery date
copyOutDate(buffer,ords-
>out_ords.item[itemCount].s_OL_DELIVERY_D_time);
appendText(&html,buffer);
appendText(&html," <BR> ");
}
//append line breaks if item count is less than 15
for (int itemCount=0;itemCount < (15-ords-
>out_ords.s_ol_cnt);itemCount++)
appendText(&html,"<BR> ");

html+=displayStatus(html,rc);

appendText(&html,"</PRE></BODY></HTML>");
return OK;
}
/*
*****
** Name : doOrderStatusErrorPage
** Description : HTML orderStatus error page
** Parameters :
** char * html page result
** char * error message
** htmlPhraser* command block
** TXN_HANDLE* txn handle
** Returns :
** int - return code
** Comments :
*****
*/
int doOrderStatusErrorPage(char *htmlPage,char
*message,htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)
{
    char *html=htmlPage;
    appendText(&html,"<HTML><HEAD><TITLE>TPC-C
Order Status</TITLE></HEAD>\n\n"
"<BODY><FORM ACTION=\n\n"
" APP_NAME
"\" METHOD=\"GET\">\n\n"
"<CENTER><H3>Please Fill In
Order Status Form.</H3></CENTER> <BR>\n\n"
"Submit Transaction <INPUT
TYPE=\n\nsubmit\" NAME=\n\n"
CMD_TXN_ID
"\" VALUE=\n\n"
CMD_ORDS

```



```

        "\>"
        "<BR> ";
html+=appendHiddenFields(html,txnHandle);

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

char buffer[15];
appendText(&html,itoa(txnHandle->w_id,buffer,10));
appendText(&html,"        District: <INPUT NAME=""
        CMD_D_ID
        \" SIZE=1>\r\n<BR>"
        "Customer: "
        "<INPUT NAME=""
        CMD_C_ID
        \" SIZE=5>"
        " "
        "Name: "
        "<INPUT NAME=""
        CMD_C_NAME
        \" SIZE=20><BR>"
        "Cust-Balance: <BR>"
        "Order-Number:      Entry-Date:

Carrier-Number<BR>"
        "Supply-W  Item-Num  Qty
Amount  Delivery <BR>");

        appendText(&html,message);
        appendText(&html,"</PRE></BODY></HTML>");
        return OK;
}
/*
*****
** Name      : doDeliveryForm
** Description : HTML payment page entry point
** Parameters : htmlPhraser*  command block
                TXN_HANDLE*  txn handle struct
** Returns   : int - return code
** Comments  :
**
*****
*/
int doDeliveryForm(htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)
{
    char *html=txnHandle->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_DLVS
        \">");

```

```

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;

    //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,com
mandBlock,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>");
        int rc =
queueDlvyTxn(dlvy.in_dlvy.s_W_ID,dlvy.in_dlvy.s_O_CARRIER
_ID);
        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[15];
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,bu
ffer,10));
appendText(&html,"<BR> <BR>Execution Status: Delivery
has been queued </PRE></BODY></HTML>");
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_DLVS
        \">");

        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;
    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,ittoa(txnHandle->w_id,buffer,10),6+1,1);
    appendText(&html,"District: ");

    appendText(&html,ittoa(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*  txn handle struct
** Returns    : int - return code
** Comments   :
*****
*/
int doStockResults(htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)
{
    char *html = txnHandle->htmlPage;
    struct stok_wrapper *stok;
    stok = (stok_wrapper*)txnHandle->comInterface.txnBuffer;
    ZeroMemory(stok,maxDataSize);
    //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,co
mmandBlock,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 = -1;

    DEBUGMSG("Calling com entry api for stock call,
w_id:"<<stok->in_stok.s_W_ID<<" d_id:"<<stok-
>in_stok.s_D_ID<<
" threshold:"<<stok->in_stok.s_threshold<<endl);
    HRESULT hres;
    try
    {
        hres = txnHandle->comInterface.comHandle-
>doStockLevel(&txnHandle-
>comInterface.size,(UCHAR**)&txnHandle-
>comInterface.txnBuffer);
    }
    catch(...)
    {
        html+=sprintf(html,"ERROR: Com Stock call caused
exeception to occur.</PRE></BODY></HTML>");
    }
}

```

```

        ERRORMSG("ERROR : Com Stock call caused
exeception to occur."<<endl);
        return OK;
    }
    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);
        return OK;
    }
}

    hres = txnHandle->comInterface.comHandle-
>doSetComplete();
    if(FAILED(hres))
    {
        html+=sprintf(html,"ERROR: stok com
doSetComplete failed, rc:%ld</PRE></BODY></HTML>",hres);
        ERRORMSG("ERROR : stok com doSetComplete
failed, rc:"<<DEBUGADDRESS(hres)<<endl);
        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>");
        ERRORMSG("ERROR stok txn failed"<<endl
        <<"stok->in_stok.s_D_ID:"<<stok-
>in_stok.s_D_ID<<endl
        <<"stok->in_stok.s_threshold:"<<stok-
>in_stok.s_threshold<<endl
        <<"stok->in_stok.s_W_ID:"<<stok-
>in_stok.s_W_ID<<endl
        <<"stok->out_stok.deadlocks:"<<stok-
>out_stok.deadlocks<<endl
        <<"stok->out_stok.s_low_stock:"<<stok-
>out_stok.s_low_stock<<endl
        <<"stok->out_stok.s_transtatus:"<<stok-
>out_stok.s_transtatus<<endl);
        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,ittoa(stok-
>in_stok.s_W_ID,buffer,10),6+1,1);
    appendText(&html,"District: ");
    appendText(&html,ittoa(stok->in_stok.s_D_ID,buffer,10));

    appendText(&html," <BR> <BR>"
              "Stock Level Threshold: ");
    appendText(&html,ittoa(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>");
        return OK;
    }
    /*
    *****
    ** Name          : doStockErrorPage
    ** Description   : HTML stock page entry point
    ** Parameters   :
    **               char *      html result page
    **               char *      query string
    **               htmlPhraser command block
    **               TXN_HANDLE * 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
**               TXN_HANDLE* txn handle struct
** 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
    appened
    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 appened
    to the html
** 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,itoa(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
    appened
    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
        \">\r\n\"
        \"<INPUT TYPE='submit\"
NAME='\"
        CMD_TXN_ID
        \" VALUE='\"
        CMD_ORDS
        \">\r\n\"
        \"<INPUT TYPE='submit\"
NAME='\"
        CMD_TXN_ID
        \" VALUE='\"
        CMD_DLVY
        \">\r\n\"
        \"<INPUT TYPE='submit\"
NAME='\"
        CMD_TXN_ID
        \" VALUE='\"
        CMD_STOK
        \">\r\n\"
        \"<INPUT TYPE='submit\"
NAME='\"
        CMD_TXN_ID
        \" VALUE='\"
        CMD_EXIT
        \">\r\n <BR>\";
    return (int)(html - htmlPage);
}
/*
*****
** Name          : appendItems
** Description   :
status page     : appends items to new order and order
** 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,itoa(cmdIDStart++,numBuffer,10));
        appendText(&html,\" SIZE=6> <INPUT NAME='\";
        appendText(&html,itoa(cmdIDStart++,numBuffer,10));
        appendText(&html,\" SIZE=6>
<INPUT NAME='\";
        appendText(&html,itoa(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);
    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);
    FILE *dlvyLog = NULL;
    char logFileName[MAX_STRING_LEN] = {NULL};
    EnterCriticalSection(&isapiLock);
    //open dlvy log file for this thread
    sprintf(logFileName,"%s\del_%d.txt",dlvyLogPath,dlvyThr
eadID);
    dlvyLog = fopen(logFileName,"w");
    if(!dlvyLog)
    {
        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;
    }
    //increment the global dlvy thread id
    dlvyThreadID++;
    LeaveCriticalSection(&isapiLock);
    DEBUGMSG("dlvyThread " << GetCurrentThreadId() << "
dlvy log file name: " << logFileName << " open." << endl);
    HANDLE workerHandles[2]; //handle
    struct DLVYQUEUEUEDATA dlvyQueueData; //dlvy queu
    struct dlvy_wrapper dlvyTxn;
    //dlvy wrapper of db2 structs
    struct _timeb endQueueTime; //time stam
    struct _timeb endProcessTime; //time sta
    char orderIDs[MAX_STRING_LEN] = {NULL}; //string to s
    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.enqueueTime.time
=
        dlvyQueue[dlvyBufferThreadIndex].enqueueTime.time;
        dlvyQueueData.enqueueTime.millitm =
        dlvyQueue[dlvyBufferThreadIndex].enqueueTime.millitm;
        dlvyQueueData.in_s_0_CARRIER_ID
=
        dlvyQueue[dlvyBufferThreadIndex].in_s_0_CARRIER_ID;
        dlvyQueueData.warehouse
=
        dlvyQueue[dlvyBufferThreadIndex].warehouse;

        DEBUGMSG("dlvyThread removed dlvy:" <<
dlvyCount << ",w_id:" << dlvyQueueData.warehouse
<< " carrier_id:" <<
dlvyQueueData.in_s_0_CARRIER_ID << endl);
        DEBUGMSG("dlvyThread removed dlvy in
queue index: " <<dlvyBufferThreadIndex<< " w_id: " <<
dlvyQueueData.warehouse
<< " carrier_id:
" << dlvyQueueData.in_s_0_CARRIER_ID << endl);
        //increment the number of free slots
        dlvyBufferFreeSlots++;
        //increment the thread index to next slot in dlvy
queue
        dlvyBufferThreadIndex++;

        DEBUGMSG("dlvyThread incremented
amount of free slots:" << dlvyBufferFreeSlots << " and thread
index:" <<
        dlvyBufferThreadIndex << endl);
        //check if we reached the end of dlvy queue, if
so, reset back index back to 0
        if(dlvyBufferThreadIndex == dlvyQueueLen)
        {
            DEBUGMSG("dlvyThread reset
dlvyBufferThreadIndex to 0, current dlvyBufferThreadIndex:" <<
dlvyBufferThreadIndex
<< " free
slots:"<<dlvyBufferFreeSlots<<endl);
            dlvyBufferThreadIndex=0;
        }

```

```

        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 << endl);

        //prepare to call database
        dlvyTxn.in_dlvy.s_0_CARRIER_ID =
        dlvyQueueData.in_s_0_CARRIER_ID;
        dlvyTxn.in_dlvy.s_W_ID =
        dlvyQueueData.warehouse;
        dlvyTxn.out_dlvy.s_transtatus = -1;

        //increment dlvy count
        dlvyCount++;
        DEBUGMSG("dlvyThread %d calling dlvy txn"
<< rc << endl);

        //call dlvy txn
        rc = dlvyCall(&dlvyTxn,connectHandle);
        _ftime(&endProcessTime);
        rc = dlvyTxn.out_dlvy.s_transtatus;

        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_0_CARRIER_ID<<
        "
        deadLocks:"<<dlvyTxn.out_dlvy.deadlocks<<" rc: "<< rc
<<endl);
        DEBUGMSG("dlvyThread dlvy s_transtatus:"
<< rc << endl);
        if(rc == OK)
        {
            bytesWritten=0;
            char *buffer = orderIDs;
            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;
            }
            else
                sprintf(orderIDs,"%nDelivery transaction
failed");

```

```

        fprintf(dlvyLog,DELIVERY_LOG_SUCCESS_STR,
                dlvyCount,
                dlvyQueueData.enqueueTime.time,
                dlvyQueueData.enqueueTime.millitm,
                endQueueTime.time,
                endQueueTime.millitm,
                dlvyQueueData.warehouse,
                dlvyQueueData.in_s_0_CARRIER_ID,
                orderIDs,
                endProcessTime.time,
                endProcessTime.millitm);

        fflush(dlvyLog);
    }
    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);
    }
} //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)
{
    DEBUGMSG("Taking lock to queue dlvy txn.");
    EnterCriticalSection(&dlvyQueueLock);
    DEBUGMSG("Lock aquired to queue dlvy txn");
    if(dlvyBufferFreeSlots)
    {
        DEBUGMSG("Checking if we are inserting at tail of
dlvy queue."<<endl);

```

```

1)    if( dlvyBufferSlotIndex == (dlvyBufferThreadIndex-
    {
        ERRORMSG("Error dlvy queue inserting over
unserviced queued dlvy txn."<<endl);
        DEBUGMSG("Error dlvy queue inserting over
unserviced queued dlvy txn."<<endl);
        LeaveCriticalSection(&dlvyQueueLock);
        return ERR_DLVS_QUEUE_EATING_TAIL;
    }
    DEBUGMSG("free slots dlvy
queue:"<<dlvyBufferFreeSlots<<" inserting txn in slot: "
<<dlvyBufferSlotIndex<<
        "w_id: "<<warehouse<<" carrier:
"<<carrier_id<<endl);
        dlvyQueue[dlvyBufferSlotIndex].warehouse =
warehouse;

        dlvyQueue[dlvyBufferSlotIndex].in_s_0_CARRIER_ID =
carrier_id;

        _ftime(&dlvyQueue[dlvyBufferSlotIndex].enqueueTime);
        //decrement the number of free slots in the buffer
dlvyBufferFreeSlots--;
        //increment the index to the next dlvy queue slot.
dlvyBufferSlotIndex++;
        DEBUGMSG("dlvy txn queued, slots available in
queue:"<<dlvyBufferFreeSlots<<" queue slot
index:"<<dlvyBufferSlotIndex
                <<"w_id:"<<warehouse<<"
carrier:"<<carrier_id<<endl);
        DEBUGMSG("dlvy txn queued, slots available in
queue: "<<dlvyBufferFreeSlots<<" queue slot index:
"<<dlvyBufferSlotIndex
                <<" w_id: "<<warehouse<<" carrier:
"<<carrier_id<<endl);
        if(dlvyBufferSlotIndex == dlvyQueueLen)
        {
            DEBUGMSG("queue slot index hit end of
queue, reset to 0, current index:"<<dlvyBufferSlotIndex<<" free
slots:"<<dlvyBufferFreeSlots<<endl);
            dlvyBufferSlotIndex=0;
        }
    }
    else
    {
        //no slots available in dlvy buffer, release critical
section and return an nord->in_nord.in_item
        LeaveCriticalSection(&dlvyQueueLock);
        ERRORMSG("dlvy queue buffer full, increase the
dlvy queue length."<<endl);
        return ERR_DLVS_QUEUE_FULL;
    }
    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);
    htmlPhaser  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 terminal 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()](&command
Block,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)
{
    //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.");
        ERRORMSG("ERROR : Terminal id given points to
a not in use 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)
{
    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;
}

```

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 - 2004
## 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 -qcpluscmt -DSQLUNIX -
DSQLAIX -q64 -O3 -D_LARGE_FILES
CFLAGS_OUT=-o
CFLAGS_DEBUG=
# Linker Configuration
LD_EXEC=xlc
LD_STORP=xlc
LDFLAGS_EXEC=-q64
LDFLAGS_SHLIB=-qmkshrobj
LDFLAGS_STORP=$(LDFLAGS_SHLIB) -bE:$@.exp -lc -b64
LDFLAGS_LIB=-L$(TPCC_SQLLIB)/lib -ldb2
LDFLAGS_OUT=-o
# Library Configuration
AR=ar
ARFLAGS=-r -v -X64
ARFLAGS_LIB=
ARFLAGS_OUT=
# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp
# OS File Extensions & Path Separators
OBJEXT=.o
LIBEXT=.a
SHLIBEXT=.a
BINEXT=
SLASH=/
CMDSEP=;

```

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 - 2004
## 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=CK040318
# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}

```

```

# The OS being used (i.e. "UNIX", "WINDOWS")
export 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=v8
# The schema name is typically the SQL authorization ID (or
username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}
# DB2 EE/EEE Configuration
export DB2EDITION=EE
#export DB2EDITION=EEE
export DB2NODE=0
export DB2NODES=1; # set to the number of nodes you
have. Set to 1 for EE.
# TPCC General Configuration
export TPCC_DBNAME=TPCC
export TPCC_ROOT=${HOME}/tpc-c.ibm
export TPCC_SQLLIB=${HOME}/sqllib
export TPCC_RUNDATA=${HOME}/tpccdata
# TPCC Debug Configuration
# This is the path where all error and debug logs are placed.
# To get debugging from within the stored procedures, you must
# set DB2ENVLIST="TPCC_DEBUGDIR" in tpcc.config.
export TPCC_DEBUGDIR=/tmp
# Specifies where stored procedures should be placed and if
they should
# be fenced.
export TPCC_SPDIR=${TPCC_SQLLIB}/function
export TPCC_FENCED=NO

```

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 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****/
/*
* db2tpcc.h - Macros and Miscellany

```

```

*/
#ifndef __DB2TPCC_H
#define __DB2TPCC_H
#include <sys/types.h>
#include "lval.h"
/*
*****
*/
/* Transaction Return Codes (s_transtatus)
*/
/*
*****
*/
#define INVALID_ITEM      100
#define TRAN_OK           0
#define FATAL_SQLERROR   -1
/*
*****
*/
/* Definition of Unused and Bad Items
*/
/*
*****
*/
/* Define unused item ID to be 0. This allows the SUT to
determine the
*/
/* number of items in the order as required by 2.4.1.3 and
2.4.2.2 since
*/
/* the assumption that any item with OL_I_ID = 0 is unused will
be true.
*/
/* This in turn requires that the value used for an invalid item is
*/
/* equal to ITEMS + 1.
*/
/*
*****
*/
#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0
#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES
/*
*****
*/
/* NURand Constants
*/
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to
clause 2.1.6.
*/
/* Analysis indicates that a C_LAST delta of 85 is optimal.
*/
/*
*****
*/
#define C_C_LAST_RUN      88
#define C_C_LAST_LOAD    173
#define C_C_ID            319
#define C_OL_I_ID        3849
#define A_C_LAST          255
#define A_C_ID            1023
#define A_OL_I_ID        8191
/*
*****
*/
/* Transaction Type Identifiers
*/
/*
*****
*/
#define CLIENT_SQL        0
#define NEWORD_SQL        1

```

```

#define PAYMENT_SQL 2
#define ORDSTAT_SQL 3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5
#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)
struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    int64_t s_O_ENTRY_D_time; /* init by SUT */
    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 {
        int32_t s_I_PRICE;
        int32_t s_OL_AMOUNT;
        int16_t s_S_QUANTITY;
        int16_t pad2;
        char s_I_NAME[25];
        char s_brand_generic;
    } item[15];
    int64_t s_O_ENTRY_D_time;
    int32_t s_W_TAX;
    int32_t s_D_TAX;
    int32_t s_C_DISCOUNT;
    int32_t 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];
};
struct in_payment_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_H_DATE_time; /* init by SUT */
    int64_t 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];
    int64_t s_H_DATE_time;

```

```

int64_t s_C_SINCE_time;
int64_t s_C_CREDIT_LIM;
int64_t s_C_BALANCE;
int32_t s_C_DISCOUNT;
int32_t s_C_ID;
int16_t s_transtatus;
int16_t deadlocks;
char s_W_STREET_1[21];
char s_W_STREET_2[21];
char s_W_CITY[21];
char s_W_STATE[3];
char s_W_ZIP[10];
char s_D_STREET_1[21];
char s_D_STREET_2[21];
char s_D_CITY[21];
char s_D_STATE[3];
char s_D_ZIP[10];
char s_C_FIRST[17];
char s_C_MIDDLE[3];
char s_C_LAST[17];
char s_C_STREET_1[21];
char s_C_STREET_2[21];
char s_C_CITY[21];
char s_C_STATE[3];
char s_C_ZIP[10];
char s_C_PHONE[17];
char s_C_CREDIT[3];
char s_C_DATA[201];
};
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];
    int64_t s_C_BALANCE;
    int64_t s_O_ENTRY_D_time;
    int32_t s_C_ID;

    int32_t s_O_ID;
    int16_t s_O_CARRIER_ID;
    int16_t s_ol_cnt;
    int16_t pad1[2];
    struct oitems_struct {
        int64_t s_OL_DELIVERY_D_time;
        int32_t s_OL_AMOUNT;
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad2;
    } item[15];
    int16_t s_transtatus;
    int16_t deadlocks;
    char s_C_FIRST[17];
    char s_C_MIDDLE[3];
    char s_C_LAST[17];
};

```



```

};
struct in_delivery_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    int64_t s_O_DELIVERY_D_time; /* init by SUT */
    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

```

#ifdef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 256000
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900
#endif // __LVAL_H

```

include/tpccapp.h

```

/*****
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
** 1996 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
** IBM Corp.
*****/
/*
* tpccapp.h - Application Macros
*/
#ifdef __TPCCAPP_H
#define __TPCCAPP_H
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
#define daricall
#include "sqlca.h"
#include "sqlcodes.h"
#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var) SwapEndian((void*)&Var,
sizeof(Var))
/*****

```

```

FUNCTION: SwapEndian
PURPOSE: Swap the byte order of a structure
EXAMPLE: int l=0x12345678; SWAP_BYTE(l); l ==>
0x78563412;
IMPLEMENTATION: Fold Addr in half, swap header & tail by
XOR op
e.g.: *a = 0x12 [ Addr + 0];
*b = 0x78 [ 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 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
** IBM Corp.

```

```

*****
****/
/*
 * tpcdbg.h - Debugging Macros
 */
#ifndef __TPCCDBG_H
#define __TPCCDBG_H
#ifdef __cplusplus
extern "C" {
#endif
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

```

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 - 2004
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication
or
## disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
#####
#####
#
# Makefile - Makefile for Src.Common
#
include $(TPCC_ROOT)/Makefile.config
#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####
BND_OPTS = GRANT PUBLIC \
            MESSAGES $.bnd.msg
PRP_OPTS = BINDFILE \
            OPTLEVEL 1 \
            ISOLATION RR \
            MESSAGES $.prep.msg \
            LEVEL $(TPCC_VERSION) \
            NOLINEMACRO
INCLUDE = -I$(TPCC_SQLLIB)/include -
          I$(TPCC_ROOT)/include
CFLAGS = $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDE) \
          -DSQLA_NOLINES -D$(DB2EDITION) -
          D$(DB2VERSION) \
          -D$(TPCC_SPTYPE)
UTIL_OBJ = tpcmisc$(OBJEXT) tpcdbg$(OBJEXT)
tpccctx$(OBJEXT)
#
#####
#####
# User Targets
#
#####
#####
all: connect $(UTIL_OBJ) disconnect
clean:
      - $(ERASE) *$(OBJEXT) *.bnd *.msg tpcctx.c
#
#####
#####
# Helper Targets
#
#####
#####
connect:
      - db2 connect to $(TPCC_DBNAME)
disconnect:
      - db2 connect reset
      - db2 terminate
rebind: connect
      db2 bind tpcctx.bnd $(BND_OPTS)

```

```

#
#####
#####
# Build Rules
#
#####
#####
.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc
.sqc.c:
      @echo "Prepping $.sqc"
      -db2 prep $.sqc $(PRP_OPTS)
      @echo "Binding $.bnd"
      db2 bind $.bnd $(BND_OPTS)
#
#####
#####
# Dependencies
#
#####
#####
# Source
tpccdbg$(OBJEXT): tpcdbg.c
tpccctx$(OBJEXT): tpcctx.c
tpccmisc$(OBJEXT): tpcmisc.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 - 2004
** 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 <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>
#include "db2tpcc.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;
    strncpy(dbname,in_dbname,8);
    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 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.

```

```

*****
****/
/*
 * tcpcdbg.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 */
sqlainp(errStr, 512, 78, psqlca);
err_fp = fopen(err_fn, "a+");
fprintf(err_fp, "-----\n");
fprintf(err_fp, "Transaction: %s (%s)\n", tranName, msg);
fprintf(err_fp, "FILE %s (%u)\n", file, line);
fprintf(err_fp, "SQLCODE %d ", psqlca->sqlcode);
fprintf(err_fp, "PID %d ", getpid());
fprintf(err_fp, "TIME %s\n", timeStamp);
fprintf(err_fp, "-----\n");
fprintf(err_fp, "%s", errStr);
fprintf(err_fp, "-----\n");
if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
{
    fprintf(err_fp, "slerrmc: ");
    for(j = 0; j < 5; j++)
    {
        for(k = 0; k < 16; k++) {
            int pos = j * 16 + k;
            if (pos < 70) fprintf(err_fp, "%02x ", psqlca-
            >sqlerrmc[pos]);
            else fprintf(err_fp, " ");
        }
        fprintf(err_fp, " |");
        for(k = 0; k < 16; k++) {
            int pos = j * 16 + k;
            char c = ' ';
            if (pos < 70) {
                c = psqlca->sqlerrmc[pos];
                if (!isprint(c)) c = ' ';
            }
            fprintf(err_fp, "%c", c);
        }
        fprintf(err_fp, "\n");
        if (j < 4) fprintf(err_fp, " ");
    }
}

```

```

    }
}
fprintf(terr_fp, "sqlerrp: ");
for(j = 0; j < 8; j++)
    fprintf(terr_fp, "%c", psqlca->sqlerrp[j]);
fprintf(terr_fp, "\n");
fprintf(terr_fp, "sqlerrd: ");
for(j = 0; j < 6; j++)
    fprintf(terr_fp, "%d", psqlca->sqlerrd[j]);
fprintf(terr_fp, "\n");
if (psqlca->sqlwarn[0] != ' ')
{
    fprintf(terr_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
        fprintf(terr_fp, "%c ", psqlca->sqlwarn[j]);
    fprintf(terr_fp, "\n");
}
fprintf(terr_fp, "\n");
fclose(terr_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, "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, "ts_O_DELIVERY_D = %lld (%lX)\n",
            in_delivery->s_O_DELIVERY_D_time, in_delivery-
>s_O_DELIVERY_D_time);
    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, "\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, "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_O_ENTRY_D = %lld (%lX)\n",
            in_neword->s_O_ENTRY_D_time, in_neword-
>s_O_ENTRY_D_time);
    // fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
    // in_neword->s_transtatus, in_neword->s_transtatus);
    // fprintf(debug_fp, "\tduplicate_items= %d (%X)\n",
    // in_neword->duplicate_items, in_neword-
>duplicate_items);
    fprintf(debug_fp, "titems {\n");
    items = in_neword->s_O_OL_CNT;
    for (j=0; j<items; j++) {
        if(j != 0)
            fprintf(debug_fp, "\n");
        fprintf(debug_fp, "\tts_OL_I_ID[%d] = %d (%X)\n",
                j, in_neword->in_item[j].s_OL_I_ID, in_neword-
>in_item[j].s_OL_I_ID);
        fprintf(debug_fp, "\tts_OL_SUPPLY_W_ID[%d] = %d
(%X)\n",
                j, in_neword->in_item[j].s_OL_SUPPLY_W_ID,
in_neword->in_item[j].s_OL_SUPPLY_W_ID);
        fprintf(debug_fp, "\tts_OL_QUANTITY[%d] = %d (%X)\n",
                j, in_neword->in_item[j].s_OL_QUANTITY, in_neword-
>in_item[j].s_OL_QUANTITY);
    }
    fprintf(debug_fp, "\t)\n");
    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 = %d\n",
            neword_ptr->s_W_TAX);
    fprintf(debug_fp, "ts_D_TAX = %d\n",
            neword_ptr->s_D_TAX);
    fprintf(debug_fp, "ts_C_DISCOUNT = %d\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 = %lld (%lX)\n",
            neword_ptr->s_O_ENTRY_D_time, neword_ptr-
>s_O_ENTRY_D_time);
    fprintf(debug_fp, "ts_total_amount = %d\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",

```



```

    in_payment->s_H_DATE_time, in_payment-
>s_H_DATE_time);
    fprintf(debug_fp, "ts_C_CREDIT_LIM = %lld\n",
    payment_ptr->s_C_CREDIT_LIM);
    fprintf(debug_fp, "ts_C_DISCOUNT = %d\n",
    payment_ptr->s_C_DISCOUNT);
    fprintf(debug_fp, "ts_C_BALANCE = %lld\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 = %lld (%lX)\n",
    payment_ptr->s_C_SINCE_time, 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=====
=====");
    fprintf(debug_fp, "in_stocklev_struct {\n");
    fprintf(debug_fp, "ts_W_ID = %d (%X)\n",
    in_stocklev->s_W_ID, in_stocklev->s_W_ID);
    fprintf(debug_fp, "ts_D_ID = %d (%X)\n",
    in_stocklev->s_D_ID, in_stocklev->s_D_ID);
    fprintf(debug_fp, "ts_threshold = %d (%X)\n",
    in_stocklev->s_threshold, in_stocklev->s_threshold);
    fprintf(debug_fp, ")\n\n");
    fprintf(debug_fp, "out_stocklev_struct {\n");
    fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
    stocklev->s_transtatus, stocklev->s_transtatus);
    fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
    stocklev->deadlocks, stocklev->deadlocks);
    fprintf(debug_fp, "ts_low_stock = %d (%X)\n",
    stocklev->s_low_stock, stocklev->s_low_stock);
    fprintf(debug_fp, ")\n\n");
    fclose(debug_fp);
}
void current_tmstamp(char *buf)
{
    time_t t = time(NULL);
    strncpy(buf, ctime(&t), 19);
}

```

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 - 2004
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
*****
****/
/*
 * tpccmisc.c - Miscellaneous routines
 */
#include <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>
double current_time_ms(void);
double current_time(void);

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

```

Src.Srv/Makefile

```

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

```

```
#####
#####
#
# Makefile - Makefile for Src.Srv
#
include $(TPCC_ROOT)/Makefile.config
#
#####
#####
# Preprocessor, Compiler and Linker Flags
#
#####
#####
BND_OPTS = GRANT PUBLIC \
            MESSAGES $*.bnd.msg
PRP_OPTS = BINDFILE \
            EXPLAIN ALL \
            MESSAGES $*.prep.msg
INCLUDE = -I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include
CFLAGS = $(CFLAGS_OS) $(INCLUDE) $(CFLAGS_DEBUG) \
         -D$(DB2EDITION) -D$(DB2VERSION) \
         -DSQLA_NOLINES -DLINT_ARGS
LD_FLAGS = $(LD_FLAGS_STORP) $(LD_FLAGS_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 uncatal-func.ddl +o -z uncatal-func.out
      - db2 -tvf uncatal-proc.ddl +o -z uncatal-proc.out
explain:
      - perl
$(TPCC_ROOT)$(SLASH)utils$(SLASH)fixup_explain.pl
      - db2 -tvf
$(TPCC_ROOT)$(SLASH)utils$(SLASH)EXPLAIN.DDL +o -z
EXPLAIN.out
unexplain:
      - db2 -tvf
$(TPCC_ROOT)$(SLASH)utils$(SLASH)UNEXPLAIN.DDL +o -
z UNEXPLAIN.out
connect:
      - db2 connect to $(TPCC_DBNAME)
disconnect:
      - db2 connect reset
      - db2 terminate
plan:
      - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -
s $(TPCC_SCHEMA) -w -1 -n TPCC_ALL -g -# 0 -o
TPCC_ALL.exfmt.plan
      - (export DB2EXPLN_BUFFER=3000000; db2expln -d
$(TPCC_DBNAME) -c $(TPCC_SCHEMA) -p TPCC_ALL -s 0 -
g -o TPCC_ALL.expln.plan )
rebind: connect catalog
      db2 bind tpcc_all_sql.bnd $(BND_OPTS) QUERYOPT 7
#
#####
#####
# Install Targets
#
#####
#####
install: $(EXE)
      - mkdir $(TPCC_SPDIR)
$(COPY) ords $(TPCC_SPDIR)
$(COPY) news $(TPCC_SPDIR)
$(COPY) dels $(TPCC_SPDIR)
#
#####
#####
# Build Rules
#
#####
#####
.SUFFIXES: $(OBJEXT) .c .sqc
# d230437mte: QUERYOPT 7 required for UNION ALL
# Only stock needs CS , and that can be specified on the
SELECT statement
tpcc_all_sql.c:
      @echo "Prepping $*.sqc"
      -db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
      @echo "Binding $*.bnd"
      db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7
# Stored procedures are built in a special way
tpcc_all_sql$(OBJEXT):
      $(CC) -c tpcc_all_sql.c $(CFLAGS) -D$(TPCC_SPTYPE)
$(CFLAGS_OUT)$(@)
```

```
$(EXE): $(UTIL_OBJ) tpcc_all_sql.o
      $(LD_STORP) $(LD_FLAGS) $(UTIL_OBJ) tpcc_all_sql.o
$(LD_FLAGS_OUT)$(@)
#
#####
#####
# 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
-- 2004
-- 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
                   , DELIVERY_D BIGINT
                   )
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 INTEGER ;
/* 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 =
DEL.DELIVERY_D
    WHERE OL_W_ID = DEL.W_ID
    AND OL_D_ID = DEL.D_ID
    AND OL_O_ID = VAR.O_ID
    ) AS U
;
/* Charge the customer */
UPDATE CUSTOMER
    SET C_BALANCE = C_BALANCE + VAR.AMOUNT
    , C_DELIVERY_CNT = C_DELIVERY_CNT +
SMALLINT( 1 )
    WHERE C_W_ID = DEL.W_ID
    AND C_D_ID = DEL.D_ID
    AND C_ID = VAR.C_ID
;
/* Return the order id to the caller (or NULL) */
RETURN VALUES VAR.O_ID ;
END
$

--
-- ORDER STATUS
--
CREATE FUNCTION ORD_C_LAST( W_ID INTEGER
    , D_ID SMALLINT
    , C_LAST VARCHAR(16)
)
RETURNS TABLE( O_ID INTEGER
    , O_CARRIER_ID SMALLINT
    , O_ENTRY_D BIGINT
    , C_BALANCE BIGINT
    , 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 BIGINT ;
DECLARE C_FIRST VARCHAR(16) ;
DECLARE C_MIDDLE CHAR(2) ;
DECLARE C_ID INTEGER ;

```

```

DECLARE O_ID INTEGER;
DECLARE O_CARRIER_ID SMALLINT;
DECLARE O_ENTRY_D BIGINT;
/* 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 )
;
/* Take advantage of the index to fetch the first row (and hence
max(o_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_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 BIGINT
    , C_BALANCE BIGINT
    , 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 BIGINT ;
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 BIGINT;
/* 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_DATE BIGINT
    , H_AMOUNT BIGINT
    , 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 BIGINT
, C_CREDIT CHAR(2)
, C_CREDIT_LIM BIGINT
, C_DISCOUNT INTEGER
, C_BALANCE BIGINT
, C_DATA CHAR(200)
)
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 BIGINT;
DECLARE C_CREDIT CHAR(2);
DECLARE C_CREDIT_LIM BIGINT;
DECLARE C_DISCOUNT INTEGER;
DECLARE C_BALANCE BIGINT;
DECLARE C_DATA CHAR(200);
/* 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
, PAY_C_LAST.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
);
END
$
CREATE FUNCTION PAY_C_ID( W_ID INTEGER
, D_ID SMALLINT
, C_W_ID INTEGER
, C_D_ID SMALLINT
, C_ID INTEGER
, H_DATE BIGINT
, H_AMOUNT BIGINT
, 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 BIGINT
, C_CREDIT CHAR(2)
, C_CREDIT_LIM BIGINT
, C_DISCOUNT INTEGER
, C_BALANCE BIGINT
, C_DATA CHAR(200)
);

```

```

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 BIGINT;
DECLARE C_CREDIT CHAR(2);
DECLARE C_CREDIT_LIM BIGINT;
DECLARE C_DISCOUNT INTEGER;
DECLARE C_BALANCE BIGINT;
DECLARE C_DATA CHAR(200);

/* 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
, PAY_C_ID.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
)
;
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 INTEGER
, 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 INTEGER;
DECLARE I_NAME CHAR(24);
DECLARE I_DATA VARCHAR(50);
DECLARE OL_DIST_INFO CHAR(24);
DECLARE S_DATA VARCHAR(50);
DECLARE S_QUANTITY SMALLINT;

SET ( I_PRICE, I_NAME, I_DATA )
= ( SELECT
I_PRICE
, I_NAME
, I_DATA
FROM ITEM
WHERE ITEM.I_ID = NEW_OL_ALL.I_ID
);
SET ( OL_DIST_INFO, S_DATA, S_QUANTITY )
= ( SELECT OL_DIST_INFO
, S_DATA
, S_QUANTITY
FROM NEW TABLE ( UPDATE STOCK
INCLUDE ( OL_DIST_INFO CHAR( 24 ) )
SET S_QUANTITY = CASE WHEN
S_QUANTITY - NEW_OL_ALL.I_QTY >= 10
THEN S_QUANTITY -
NEW_OL_ALL.I_QTY
ELSE S_QUANTITY -
NEW_OL_ALL.I_QTY + 91
END
, S_ORDER_CNT =
S_ORDER_CNT + SMALLINT( 1 )
, S_YTD = S_YTD +
NEW_OL_ALL.I_QTY
, S_REMOTE_CNT = CASE
WHEN NEW_OL_ALL.SUPP_W_ID = NEW_OL_ALL.W_ID
THEN
S_REMOTE_CNT
ELSE
S_REMOTE_CNT + SMALLINT( 1 )
END
, OL_DIST_INFO = CASE
D_ID WHEN SMALLINT( 1 ) THEN S_DIST_01
WHEN
SMALLINT( 2 ) THEN S_DIST_02
WHEN
SMALLINT( 3 ) THEN S_DIST_03

```

```

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     WHEN
                                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 INTEGER
, 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 INTEGER ;
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
SMALLINT( 2 ) THEN S_DIST_02
SMALLINT( 3 ) THEN S_DIST_03
SMALLINT( 4 ) THEN S_DIST_04
SMALLINT( 5 ) THEN S_DIST_05
SMALLINT( 6 ) THEN S_DIST_06
SMALLINT( 7 ) THEN S_DIST_07
SMALLINT( 8 ) THEN S_DIST_08
SMALLINT( 9 ) THEN S_DIST_09
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_ENTRY_D BIGINT
, O_OL_CNT SMALLINT

```

```

, O_ALL_LOCAL SMALLINT
)
RETURNS TABLE ( W_TAX INTEGER
, C_DISCOUNT INTEGER
, C_LAST VARCHAR(16)
, C_CREDIT CHAR(2)
)
SPECIFIC NEW_WH
MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL
ACTION LANGUAGE SQL
VAR: BEGIN ATOMIC
DECLARE C_DISCOUNT INTEGER ;
DECLARE C_LAST VARCHAR(16) ;
DECLARE C_CREDIT CHAR(2) ;
DECLARE W_TAX INTEGER ;
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 );
END
$

```

Src.Srv/cat-proc.ddl

```

CREATE PROCEDURE news
(in new_in varchar(270) FOR BIT DATA,
out new_out varchar(662) 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(446) 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(22) 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/dels.exp

```

#! Export file
dels

```

Src.Srv/news.exp

```

#! Export file
news

```

Src.Srv/ords.exp

```

#! Export file
ords

```

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 - 2004
** 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"
#include "lval.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 ];
sqlint32 c_discount ;
sqlint32 dist_tax ;
sqlint32 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 ];
sqlint64 o_entry_d ;

short allLocal ;

sqlint32 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 o_entry_d in_neword->s_O_ENTRY_D_time
#define inputItemCount in_neword->s_O_OL_CNT
#define allLocal in_neword->s_all_local
// Redirected output fields
#define c_last neword->s_C_LAST
#define c_credit neword->s_C_CREDIT
#define c_discount neword->s_C_DISCOUNT
#define ware_tax neword->s_W_TAX
#define dist_tax neword->s_D_TAX
#define s_quantity neword-
>item[ inputItemArrayIndex ].s_S_QUANTITY
// This output field becomes an input field to order_line

#define next_o_id neword->s_O_ID

#define item_name neword-
>item[ inputItemArrayIndex ].s_I_NAME
// item_price holds the integer version of this value. If the
return structure was
// an integer this would not be necessary.
sqlint32 i_priceArray[ 15 ];
#define item_price i_priceArray[ inputItemArrayIndex ]
// Handle the generic/brand distinction
struct i_data_type i_dataArray[ 15 ];
struct s_data_type s_dataArray[ 15 ];
#define i_data i_dataArray[ inputItemArrayIndex ]
#define s_data s_dataArray[ inputItemArrayIndex ]

// Redirect hostvars to input structure
#define id0 in_neword->in_item[0].s_OL_I_ID
#define id1 in_neword->in_item[1].s_OL_I_ID
#define id2 in_neword->in_item[2].s_OL_I_ID
#define id3 in_neword->in_item[3].s_OL_I_ID
#define id4 in_neword->in_item[4].s_OL_I_ID
#define id5 in_neword->in_item[5].s_OL_I_ID
#define id6 in_neword->in_item[6].s_OL_I_ID
#define id7 in_neword->in_item[7].s_OL_I_ID
#define id8 in_neword->in_item[8].s_OL_I_ID
#define id9 in_neword->in_item[9].s_OL_I_ID
#define id10 in_neword->in_item[10].s_OL_I_ID
#define id11 in_neword->in_item[11].s_OL_I_ID
#define id12 in_neword->in_item[12].s_OL_I_ID
#define id13 in_neword->in_item[13].s_OL_I_ID
#define id14 in_neword->in_item[14].s_OL_I_ID
#define ol_quantity0 in_neword-
>in_item[ 0 ].s_OL_QUANTITY
#define ol_quantity1 in_neword-
>in_item[ 1 ].s_OL_QUANTITY
#define ol_quantity2 in_neword-
>in_item[ 2 ].s_OL_QUANTITY
#define ol_quantity3 in_neword-
>in_item[ 3 ].s_OL_QUANTITY
#define ol_quantity4 in_neword-
>in_item[ 4 ].s_OL_QUANTITY

```

```

#define ol_quantity5 in_neword-
>in_item[ 5].s_OL_QUANTITY
#define ol_quantity6 in_neword-
>in_item[ 6].s_OL_QUANTITY
#define ol_quantity7 in_neword-
>in_item[ 7].s_OL_QUANTITY
#define ol_quantity8 in_neword-
>in_item[ 8].s_OL_QUANTITY
#define ol_quantity9 in_neword-
>in_item[ 9].s_OL_QUANTITY
#define ol_quantity10 in_neword-
>in_item[ 10].s_OL_QUANTITY
#define ol_quantity11 in_neword-
>in_item[ 11].s_OL_QUANTITY
#define ol_quantity12 in_neword-
>in_item[ 12].s_OL_QUANTITY
#define ol_quantity13 in_neword-
>in_item[ 13].s_OL_QUANTITY
#define ol_quantity14 in_neword-
>in_item[ 14].s_OL_QUANTITY
#define supply_w_id0 in_neword-
>in_item[ 0].s_OL_SUPPLY_W_ID
#define supply_w_id1 in_neword-
>in_item[ 1].s_OL_SUPPLY_W_ID
#define supply_w_id2 in_neword-
>in_item[ 2].s_OL_SUPPLY_W_ID
#define supply_w_id3 in_neword-
>in_item[ 3].s_OL_SUPPLY_W_ID
#define supply_w_id4 in_neword-
>in_item[ 4].s_OL_SUPPLY_W_ID
#define supply_w_id5 in_neword-
>in_item[ 5].s_OL_SUPPLY_W_ID
#define supply_w_id6 in_neword-
>in_item[ 6].s_OL_SUPPLY_W_ID
#define supply_w_id7 in_neword-
>in_item[ 7].s_OL_SUPPLY_W_ID
#define supply_w_id8 in_neword-
>in_item[ 8].s_OL_SUPPLY_W_ID
#define supply_w_id9 in_neword-
>in_item[ 9].s_OL_SUPPLY_W_ID
#define supply_w_id10 in_neword-
>in_item[ 10].s_OL_SUPPLY_W_ID
#define supply_w_id11 in_neword-
>in_item[ 11].s_OL_SUPPLY_W_ID
#define supply_w_id12 in_neword-
>in_item[ 12].s_OL_SUPPLY_W_ID
#define supply_w_id13 in_neword-
>in_item[ 13].s_OL_SUPPLY_W_ID
#define supply_w_id14 in_neword-
>in_item[ 14].s_OL_SUPPLY_W_ID
EXEC SQL DECLARE ISOL_Remote_1 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO

```

```

, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;

```

```

EXEC SQL DECLARE ISOL_Remote_2 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
,
( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS ITEMLIST
, TABLE( NEW_OL_ALL( I_ID
, I_QTY
, W_ID
, I_SUPPLY_W_ID
, O_ID
, D_ID
)
) AS NEW_OL_ALL
WHERE NEW_OL_ALL.I_PRICE IS NOT
NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, 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
        , 0 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 INTEGER
, 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
        , 0 AS OL_DELIVERY_D
        , I_QTY
        , (I_PRICE * I_QTY) AS TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_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 INTEGER
, 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
        , 0 AS OL_DELIVERY_D
        , I_QTY
        , (I_PRICE * I_QTY) AS TOTAL_PRICE
        , OL_DIST_INFO
        , I_PRICE, I_NAME, I_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 )
I_ID , I_QTY , I_SUPPLY_W_ID )
) AS X ( OL_NUMBER ,
, 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 INTEGER
, 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
, 0 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
) 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 INTEGER
, 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
, 0 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
) 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 INTEGER
, I_NAME CHAR(24)

```

```

        )
        ) 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 INTEGER
    , 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
    , OL_DELIVERY_D
    , I_QTY
    , (I_PRICE * I_QTY ) AS TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,
    S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
    , :w_id AS W_ID
    , :d_id as D_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , I_QTY

```

```

    FROM Table( VALUES
    ( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
    ( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
    ( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
    ( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
    ( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
    ( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
    ( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
    ( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
    ) AS X ( OL_NUMBER ,
    I_ID , I_QTY , I_SUPPLY_W_ID )
    ) AS ITEMLIST
    , TABLE( NEW_OL_ALL( I_ID
    , I_QTY
    , W_ID
    , I_SUPPLY_W_ID
    , O_ID
    , D_ID
    )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT
    NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
    S_DATA , S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE
    ( OL_O_ID
    , OL_D_ID
    , OL_W_ID
    , OL_NUMBER
    , OL_I_ID
    , OL_SUPPLY_W_ID
    , OL_DELIVERY_D
    , OL_QUANTITY
    , OL_AMOUNT
    , OL_DIST_INFO
    )
    INCLUDE ( I_PRICE INTEGER
    , 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
    , OL_DELIVERY_D
    , I_QTY
    , (I_PRICE * I_QTY ) AS TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE, I_NAME, I_DATA, S_DATA,
    S_QUANTITY
    FROM ( SELECT :next_o_id as O_ID
    , :w_id AS W_ID
    , :d_id as D_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , I_QTY
    )
    FROM Table( VALUES
    ( SMALLINT( 1 ) , :id0 , :ol_quantity0 , :supply_w_id0 )
    ( SMALLINT( 2 ) , :id1 , :ol_quantity1 , :supply_w_id1 )
    ( SMALLINT( 3 ) , :id2 , :ol_quantity2 , :supply_w_id2 )
    ( SMALLINT( 4 ) , :id3 , :ol_quantity3 , :supply_w_id3 )
    ( SMALLINT( 5 ) , :id4 , :ol_quantity4 , :supply_w_id4 )
    ( SMALLINT( 6 ) , :id5 , :ol_quantity5 , :supply_w_id5 )
    ( SMALLINT( 7 ) , :id6 , :ol_quantity6 , :supply_w_id6 )
    ( SMALLINT( 8 ) , :id7 , :ol_quantity7 , :supply_w_id7 )
    ( SMALLINT( 9 ) , :id8 , :ol_quantity8 , :supply_w_id8 )
    ) AS X ( OL_NUMBER ,
    I_ID , I_QTY , I_SUPPLY_W_ID )
    ) AS ITEMLIST
    , TABLE( NEW_OL_ALL( I_ID
    , I_QTY
    , W_ID
    , I_SUPPLY_W_ID
    , O_ID
    , D_ID
    )
    ) AS NEW_OL_ALL
    WHERE NEW_OL_ALL.I_PRICE IS NOT
    NULL
    )
    SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
    S_DATA , S_QUANTITY
    FROM NEW TABLE ( INSERT INTO ORDER_LINE

```



```

( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_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 INTEGER
, 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
, 0 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 INTEGER
, 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
, 0 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_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 INTEGER
, 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

```

```

( 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 INTEGER
, 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
, 0 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 INTEGER
, 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
, O 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 INTEGER
, 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
, O 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 INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_1 CURSOR FOR

```

```

WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY
)
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT )
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID

```

```

, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_2 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 AS OL_DELIVERY_D
, I_QTY
, ( I_PRICE * I_QTY ) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1) , :id0 , :ol_quantity0 )
( SMALLINT( 2) , :id1 , :ol_quantity1 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY
)
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO

```

```

)
INCLUDE ( I_PRICE INTEGER
, 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
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

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

( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
)

```

```

SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, 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
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY

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

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

```

```

( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, 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
, 0 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 INTEGER
, 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
, 0 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 INTEGER
, 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
, 0 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 INTEGER
        , 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
        , O 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 INTEGER
        , 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
        , O 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 INTEGER
, I_NAME CHAR(24)
, I_DATA VARCHAR(50)
, S_DATA VARCHAR(50)
, S_QUANTITY SMALLINT)
SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, I_SUPPLY_W_ID
, OL_DELIVERY_D
, I_QTY
, TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA,
S_DATA, S_QUANTITY
FROM DATA
) AS INS
;
EXEC SQL DECLARE ISOL_Local_10 CURSOR FOR
WITH DATA AS ( SELECT O_ID
, D_ID
, W_ID
, OL_NUMBER
, I_ID
, W_ID AS I_SUPPLY_W_ID
, 0 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

```

```

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 INTEGER
, 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
)
FROM Table( VALUES

```

```

, 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
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )
( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY
)
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
)
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL

```



```

)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, 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
, 0 AS OL_DELIVERY_D
, I_QTY
, (I_PRICE * I_QTY) AS TOTAL_PRICE
, OL_DIST_INFO
, I_PRICE, I_NAME, I_DATA, S_DATA,
S_QUANTITY
FROM ( SELECT :next_o_id as O_ID
, :w_id AS W_ID
, :d_id as D_ID
, OL_NUMBER
, I_ID
, I_QTY
FROM Table( VALUES
( SMALLINT( 1 ) , :id0 , :ol_quantity0 )
( SMALLINT( 2 ) , :id1 , :ol_quantity1 )

```

```

( SMALLINT( 3 ) , :id2 , :ol_quantity2 )
( SMALLINT( 4 ) , :id3 , :ol_quantity3 )
( SMALLINT( 5 ) , :id4 , :ol_quantity4 )
( SMALLINT( 6 ) , :id5 , :ol_quantity5 )
( SMALLINT( 7 ) , :id6 , :ol_quantity6 )
( SMALLINT( 8 ) , :id7 , :ol_quantity7 )
( SMALLINT( 9 ) , :id8 , :ol_quantity8 )
( SMALLINT( 10 ) , :id9 , :ol_quantity9 )
( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
) AS X ( OL_NUMBER ,
I_ID , I_QTY )
) AS ITEMLIST
, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
) AS NEW_OL_LOCAL
WHERE NEW_OL_LOCAL.I_PRICE IS
NOT NULL
)
SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO ,
S_DATA , S_QUANTITY
FROM NEW TABLE ( INSERT INTO ORDER_LINE
( OL_O_ID
, OL_D_ID
, OL_W_ID
, OL_NUMBER
, OL_I_ID
, OL_SUPPLY_W_ID
, OL_DELIVERY_D
, OL_QUANTITY
, OL_AMOUNT
, OL_DIST_INFO
)
INCLUDE ( I_PRICE INTEGER
, 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
, 0 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 INTEGER
        , 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
    , 0 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 INTEGER
        , 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
    , 0 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 INTEGER
    , I_NAME CHAR(24)
    , I_DATA VARCHAR(50)
    , S_DATA VARCHAR(50)
    , S_QUANTITY SMALLINT )

    SELECT O_ID
    , D_ID
    , W_ID
    , OL_NUMBER
    , I_ID
    , I_SUPPLY_W_ID
    , OL_DELIVERY_D
    , I_QTY
    , TOTAL_PRICE
    , OL_DIST_INFO
    , I_PRICE , I_NAME , I_DATA ,
    S_DATA , S_QUANTITY
    FROM DATA
    ) AS INS
;
// Start processing
in_newword = (struct in_newword_struct *) pin ;
newword = (struct out_newword_struct *) pout ;
#ifdef DEBUGIT
    new_debug( newword , in_newword , "SP upon entry");
#endif
// Using I_PRICE == 0 as a flag to the client that the ITEM
was not fetched (hence bad).

```

```

    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
    in_newword->s_O_OL_CNT ; inputItemArrayIndex++ )
    {
        i_priceArray[ inputItemArrayIndex ] = 0 ;
    }
    newword->deadlocks = -1 ;
    retry_tran:

    newword->deadlocks++ ;
    EXEC SQL
    SELECT D_TAX , D_NEXT_O_ID
    INTO :dist_tax , :next_o_id
    FROM OLD TABLE ( UPDATE DISTRICT
    SET D_NEXT_O_ID = D_NEXT_O_ID +
    1
    WHERE D_W_ID = :w_id
    AND D_ID = :d_id
    ) AS OT
;
if ( sqlca.sqlcode != 0 )
{
    DLCHK( retry_tran );
    sqlerror( NEWWORD_SQL , "DISTRICT" , __FILE__ ,
    __LINE__ , &sqlca ) ;
    goto ferror ;
}
// Invalid I_ID will give a +100, now that we've changed the
cursor definitions
// to include a 'WHERE I_PRICE NOT NULL' clause.
#define NEW_CURSOR_OPEN_ERROR
\
{
    if( sqlca.sqlcode != 0 )
    {
        goto sql_error ;
    }
}
#define NEW_CURSOR_ERROR
\
{
    if( sqlca.sqlcode == 0 )
    {
        newword->s_O_OL_CNT ++ ;
    }
    else
    if( sqlca.sqlcode == +100 )
    {
        break ;
    }
    else
    goto sql_error ;
}
if ( allLocal )
{
    switch( inputItemCount )
    {
        case 1:
            EXEC SQL OPEN ISOL_Local_1 ;
            NEW_CURSOR_OPEN_ERROR
            for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
            inputItemCount ; inputItemArrayIndex++ )

```

```

    {
        EXEC SQL FETCH ISOL_Local_1
        INTO :item_price , :item_name , :i_data , :stockDistrictInformation
        , :s_data , :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 2:
    EXEC SQL OPEN ISOL_Local_2 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
    inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_2
        INTO :item_price , :item_name , :i_data , :stockDistrictInformation
        , :s_data , :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 3:
    EXEC SQL OPEN ISOL_Local_3 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
    inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_3
        INTO :item_price , :item_name , :i_data , :stockDistrictInformation
        , :s_data , :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 4:
    EXEC SQL OPEN ISOL_Local_4 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
    inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_4
        INTO :item_price , :item_name , :i_data , :stockDistrictInformation
        , :s_data , :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 5:
    EXEC SQL OPEN ISOL_Local_5 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
    inputItemCount ; inputItemArrayIndex++ )
    {
        EXEC SQL FETCH ISOL_Local_5
        INTO :item_price , :item_name , :i_data , :stockDistrictInformation
        , :s_data , :s_quantity ;
        NEW_CURSOR_ERROR
    }
    break ;
case 6:
    EXEC SQL OPEN ISOL_Local_6 ;
    NEW_CURSOR_OPEN_ERROR
    for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
    inputItemCount ; inputItemArrayIndex++ )
    {

```

```

EXEC SQL FETCH ISOL_Local_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 7:
EXEC SQL OPEN ISOL_Local_7;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 8:
EXEC SQL OPEN ISOL_Local_8;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 9:
EXEC SQL OPEN ISOL_Local_9;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 10:
EXEC SQL OPEN ISOL_Local_10;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 11:
EXEC SQL OPEN ISOL_Local_11;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{

```

```

EXEC SQL FETCH ISOL_Local_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 12:
EXEC SQL OPEN ISOL_Local_12;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 13:
EXEC SQL OPEN ISOL_Local_13;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 14:
EXEC SQL OPEN ISOL_Local_14;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 15:
EXEC SQL OPEN ISOL_Local_15;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Local_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
default:
sqlerror(NEWORD_SQL, "Default switch on local
orderline/stock/index", __FILE__, __LINE__, &sqlca);
goto ferror;
}
}
else

```

```

{
switch( inputItemCount )
{
case 1:
EXEC SQL OPEN ISOL_Remote_1;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Remote_1
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 2:
EXEC SQL OPEN ISOL_Remote_2;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Remote_2
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 3:
EXEC SQL OPEN ISOL_Remote_3;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Remote_3
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 4:
EXEC SQL OPEN ISOL_Remote_4;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Remote_4
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
break;
case 5:
EXEC SQL OPEN ISOL_Remote_5;
NEW_CURSOR_OPEN_ERROR
for ( inputItemArrayIndex = 0; inputItemArrayIndex <
inputItemCount; inputItemArrayIndex++)
{
EXEC SQL FETCH ISOL_Remote_5
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data, :s_quantity;
NEW_CURSOR_ERROR
}
}
}
}

```

```

        break ;
    case 6:
        EXEC SQL OPEN ISOL_Remote_6 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_6
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 7:
        EXEC SQL OPEN ISOL_Remote_7 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_7
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 8:
        EXEC SQL OPEN ISOL_Remote_8 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_8
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 9:
        EXEC SQL OPEN ISOL_Remote_9 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_9
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 10:
        EXEC SQL OPEN ISOL_Remote_10 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_10
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 11:

```

```

        EXEC SQL OPEN ISOL_Remote_11 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_11
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 12:
        EXEC SQL OPEN ISOL_Remote_12 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_12
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 13:
        EXEC SQL OPEN ISOL_Remote_13 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_13
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 14:
        EXEC SQL OPEN ISOL_Remote_14 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_14
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    case 15:
        EXEC SQL OPEN ISOL_Remote_15 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex <
inputItemCount ; inputItemArrayIndex++ )
        {
            EXEC SQL FETCH ISOL_Remote_15
INTO :item_price, :item_name, :i_data, :stockDistrictInformation
, :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
        }
        break ;
    default:

```

```

        sqlerror(NEWORD_SQL, "Default switch on remote
orderline/stock/index", __FILE__, __LINE__, &sqlca);
        goto ferror;
    }
}
for ( inputItemArrayIndex = 0 ;
inputItemArrayIndex < in_neword->s_O_OL_CNT //
from input
&& i_priceArray[ inputItemArrayIndex ] != 0 ;
inputItemArrayIndex++ )
{
    // s_I_NAME, and s_S_QUANTITY already set as output
host variables
    neword->item[ inputItemArrayIndex ].s_I_PRICE =
i_priceArray[ inputItemArrayIndex ] ;

    if
( is_ORIGINAL( s_dataArray[ inputItemArrayIndex ].data,
s_dataArray[ inputItemArrayIndex ].len )
&&
is_ORIGINAL( i_dataArray[ inputItemArrayIndex ].data,
i_dataArray[ inputItemArrayIndex ].len ) )
    {
        neword->item[ inputItemArrayIndex ].s_brand_generic =
'B';
    }
    else
    {
        neword->item[ inputItemArrayIndex ].s_brand_generic =
'G';
    }
}
EXEC SQL
SELECT W_TAX, C_DISCOUNT, C_LAST, C_CREDIT
INTO :ware_tax, :c_discount, :c_last, :c_credit

FROM TABLE ( NEW_WH ( :next_o_id
, :w_id
, :d_id
, :c_id
, :o_entry_d
, :inputItemCount
, :allLocal
) AS NEW_WH_TABLE
;
if ( sqlca.sqlcode == 0 )
{
    if ( neword->s_O_OL_CNT == in_neword->s_O_OL_CNT )
    {
        neword->s_transtatus = TRAN_OK ;
        EXEC SQL COMMIT;

        if( sqlca.sqlcode != 0 )
        {
            sqlerror(NEWORD_SQL, "COMMIT", __FILE__,
__LINE__, &sqlca ) ;
            goto ferror;
        }
    }
    else
    {

```

```

neword->s_transtatus = INVALID_ITEM ;

EXEC SQL ROLLBACK WORK ;

if ( sqlca.sqlcode != 0 )
{
    neword->s_transtatus = FATAL_SQLERROR;
    sqlerror(NEWORD_SQL, "ROLLBACK FAILED
(INVALID ITEM)", __FILE__, __LINE__, &sqlca);
    // no point in ferror
}
}
else
{
    DLCHK( retry_tran );
    sqlerror( NEWORD_SQL, "NEW_WH", __FILE__,
__LINE__, &sqlca);
    goto ferror;
}
/*-----*/
/* Return to client */
/*-----*/
mexit:
if ( sqlca.sqlcode >= 0 )
{
    storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
    storedProcRc = SQLZ_DISCONNECT_PROC ;
}
#ifdef DEBUGIT
new_debug( neword, in_neword, "SP prior to return");
#endif
return ( storedProcRc );
sql_error:
{
    char tempstr[ 4096 ];
    DLCHK( retry_tran );
    sprintf( tempstr,
"inputItemCnt=%d, :next_o_id=%d, :d_id=%d, :w_id=%d",
inputItemCnt, next_o_id, d_id, w_id );
    sqlerror( NEWORD_SQL, tempstr, __FILE__, __LINE__,
&sqlca );
}
ferror:
neword->s_transtatus = FATAL_SQLERROR;
EXEC SQL ROLLBACK WORK;
if ( sqlca.sqlcode != 0 )
{
    sqlerror( NEWORD_SQL, "ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca );
}

goto mexit ;
}
/*
** A little function to search for the string "ORIGINAL" given a
string and
** it's length
*/

```

```

static unsigned char skip[256] = {8,8,8,8,8,8,8,8, /*0-9*/
    8,8,8,8,8,8,8,8, /*10-19*/
    8,8,8,8,8,8,8,8, /*20-29*/
    8,8,8,8,8,8,8,8, /*30-39*/
    8,8,8,8,8,8,8,8, /*40-49*/
    8,8,8,8,8,8,8,8, /*50-59*/
    8,8,8,8,1,8,8,8,8, /*60-69*/
    8,4,8,3,8,8,0,8,2,7, /*70-79*/
    8,8,6,8,8,8,8,8,8,8, /*80-89*/
    8,8,8,8,8,8,8,8,8,8, /*90-99*/
    8,8,8,8,8,8,8,8,8,8, /*100-109*/
    8,8,8,8,8,8,8,8,8,8, /*110-119*/
    8,8,8,8,8,8,8,8,8,8, /*120-129*/
    8,8,8,8,8,8,8,8,8,8, /*130-139*/
    8,8,8,8,8,8,8,8,8,8, /*140-149*/
    8,8,8,8,8,8,8,8,8,8, /*150-159*/
    8,8,8,8,8,8,8,8,8,8, /*160-169*/
    8,8,8,8,8,8,8,8,8,8, /*170-179*/
    8,8,8,8,8,8,8,8,8,8, /*180-189*/
    8,8,8,8,8,8,8,8,8,8, /*190-199*/
    8,8,8,8,8,8,8,8,8,8, /*200-209*/
    8,8,8,8,8,8,8,8,8,8, /*210-219*/
    8,8,8,8,8,8,8,8,8,8, /*220-229*/
    8,8,8,8,8,8,8,8,8,8, /*230-239*/
    8,8,8,8,8,8,8,8,8,8, /*240-249*/
    8,8,8,8,8,8,8,8,8,8, /*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 ] ;
    sqlint64 c_balance ;
    // From cursor
    sqlint32 ol_i_id ;
    sqlint32 ol_supply_w_id ;
    short ol_quantity ;
    sqlint32 ol_amount ;
    sqlint64 ol_delivery_d ;
    EXEC SQL END DECLARE SECTION;
    // NOTE: this varchar would normally live inside the declare
section
    // but this package already delcared the same field higher up.
Need the field
    // within this scope though.
    ##struct s_data_type { short len ; char data[ 16 ] ; }
c_last_input ;
    int storedProcRc ;
    int itemArrayIndex = 0 ;
    #define w_id in_ordstat->s_W_ID ;
    #define d_id in_ordstat->s_D_ID ;
    #define c_id_input in_ordstat->s_C_ID
    #define o_id ordstat->s_O_ID
    #define o_entry_d ordstat->s_O_ENTRY_D_time
    #define o_carrier_id ordstat->s_O_CARRIER_ID
    #define c_id ordstat->s_C_ID
    #define c_first ordstat->s_C_FIRST
    #define c_middle ordstat->s_C_MIDDLE
    #define c_last ordstat->s_C_LAST
    #define c_balance ordstat->s_C_BALANCE
    EXEC SQL DECLARE read_orderline_cur CURSOR FOR

```

```

SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY,
OL_AMOUNT, OL_DELIVERY_D
FROM ORDER_LINE
WHERE OL_W_ID = :w_id
AND OL_D_ID = :d_id
AND OL_O_ID = :o_id
FOR FETCH ONLY ;
ordstat->deadlocks = -1 ;
#ifdef DEBUGIT
ord_debug(ordstat, in_ordstat, "SP upon entry");
#endif
retry_tran:
ordstat->deadlocks ++ ;
if ( c_id_input == 0 )
{
c_last_input.len = strlen( in_ordstat->s_C_LAST );
memcpy( c_last_input.data, in_ordstat->s_C_LAST,
c_last_input.len );
EXEC SQL
SELECT O_ID, O_CARRIER_ID, O_ENTRY_D,
C_BALANCE, C_FIRST, C_MIDDLE, C_ID
INTO :o_id, :o_carrier_id, :o_entry_d, :c_balance, :c_first, :c_m
iddle, :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_m
iddle, :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 ;
ordstat-
>item[ itemArrayIndex ].s_OL_DELIVERY_D_time =
ol_delivery_d ;

itemArrayIndex++;
}
else
if ( sqlca.sqlcode < 0 )
{
DLCHK( retry_tran );
sqlerror( ORDSTAT_SQL, "FETCH CURSOR
read_orderline_cur", __FILE__, __LINE__, &sqlca );
goto ferror ;
}
}
while ( sqlca.sqlcode == 0 ) ;
}
ordstat->s_ol_cnt = itemArrayIndex ;
EXEC SQL COMMIT ;
if ( sqlca.sqlcode == 0 )
{
ordstat->s_transtatus = TRAN_OK ;
}
else
{
DLCHK( retry_tran );
sqlerror(ORDSTAT_SQL, "COMMIT", __FILE__,
__LINE__, &sqlca);
goto ferror ;
}
mexit:
if ( sqlca.sqlcode >= 0 )
{
storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
storedProcRc = SQLZ_DISCONNECT_PROC ;
}
#ifdef DEBUGIT

```

```

ord_debug(ordstat, in_ordstat, "SP prior to return");
#endif
return ( storedProcRc );
ferror:
ordstat->s_transtatus = FATAL_SQLERROR ;
EXEC SQL ROLLBACK WORK ;
if ( sqlca.sqlcode != 0 )
{
sqlerror(ORDSTAT_SQL, "ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca);
}
goto mexit;
}
//-----
// Delivery SERVER
//-----

#ifdef d_id
#ifdef c_id
#ifdef w_id
#ifdef o_carrier_id
#ifdef 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 ;
#endif
#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 , :ol_delivery_d ) ) AS T ;

        COMMIT ;
        END COMPOUND ;

if ( sqlca.sqlcode == 0 )
{
/* Refer to clause 2.7.4.2, bullet 3 in spec.*/
/* Need to report if more than 1 or 1% of */
/* no_o_id will remain 0 if null returned, so just treat the
same way */
delivery->s_O_ID[ d_id - 1 ] = no_o_id ;
}
else
{
DLCHK( retry_tran );

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

```

```

delivery->s_transtatus = TRAN_OK ;
mexit:
if ( sqlca.sqlcode >= 0 )
{
storedProcRc = SQLZ_HOLD_PROC ;
}
else
{
storedProcRc = SQLZ_DISCONNECT_PROC ;
}
#ifdef DEBUGIT
del_debug( delivery, in_delivery, "SP prior to return");
#endif
return ( storedProcRc ) ;
ferror:
delivery->s_transtatus = FATAL_SQLERROR ;
EXEC SQL ROLLBACK WORK ;
if ( sqlca.sqlcode != 0 )
{
sqlerror( DELIVERY_SQL, "ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca ) ;
}

goto mexit ;
}
// -----
// Stored Procedure Stubs
// -----
SQL_API_RC SQL_API_FN news( char *pin, char *pout )
{
return new_order_internal( pin, pout ) ;
}
SQL_API_RC SQL_API_FN ords( char *pin, char *pout )
{
return order_status_internal( pin, pout ) ;
}
SQL_API_RC SQL_API_FN dels ( char * pin, char * pout )
{
return delivery_internal( pin, pout ) ;
}

```

Src.Srv/uncat func.ddl

```

-----
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996
-- 2004
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
-----
-- uncat-func.ddl - Drop table function DDL
--
-- STOCK LEVEL

```

```

DROP SPECIFIC FUNCTION STOCK_LEVEL $
-- DELIVERY
DROP SPECIFIC FUNCTION DELIVERY $
-- ORDER STATUS
DROP SPECIFIC FUNCTION ORD_C_LAST $
DROP SPECIFIC FUNCTION ORD_C_ID $
-- PAYMENT
DROP SPECIFIC FUNCTION PAY_C_LAST $
DROP SPECIFIC FUNCTION PAY_C_ID $
-- NEW ORDER
DROP SPECIFIC FUNCTION NEW_OL_ALL $
DROP SPECIFIC FUNCTION NEW_OL_LOCAL $
DROP SPECIFIC FUNCTION NEW_WH $

```

Src.Srv/uncat proc.ddl

```

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

DROP PROCEDURE pays;
DROP PROCEDURE ords
        (varchar(42),varchar(446));
DROP PROCEDURE ords;
DROP PROCEDURE dels
        (varchar(22),varchar(50));
DROP PROCEDURE dels;
DROP PROCEDURE stks
        (varchar(18),varchar(14));
DROP PROCEDURE stks;

```

utils/EXPLAIN.ddl

```

-- *- sql *-
--
-- Sample DDL to create Explain tables for Version 5.0
--
-- -> assumes db2start issued
-- -> assumes connection to a database exists
-- -> assumes called by "db2 -tf EXPLAIN.DDL"
--
--
-- To remind users how to use this file!
--
ECHO          ;
ECHO ***** IMPORTANT ***** ;
ECHO          ;
ECHO USAGE: db2 -tf EXPLAIN.DDL ;
ECHO          ;
ECHO ***** IMPORTANT ***** ;
ECHO          ;
ECHO          ;
--
--
-- Set autocommit off
--
UPDATE COMMAND OPTIONS USING C OFF;
--
-- EXPLAIN INSTANCE
--
-- (must be defined first due to referential integrity defintions)

```



```

--
CREATE TABLE EXPLAIN_INSTANCE
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA   VARCHAR(128)
NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
NOT NULL,
  EXPLAIN_OPTION  CHAR(1) NOT
NULL,
  SNAPSHOT_TAKEN CHAR(1) NOT
NULL,
  DB2_VERSION    CHAR(7) NOT NULL,
  SQL_TYPE       CHAR(1) NOT NULL,
  QUERYOPT       INTEGER NOT NULL,
  BLOCK          CHAR(1) NOT NULL,
  ISOLATION      CHAR(2) NOT NULL,
  BUFFPAGE      INTEGER NOT NULL,
  AVG_APPLS     INTEGER NOT NULL,
  SORTHEAP      INTEGER NOT NULL,
  LOCKLIST      INTEGER NOT NULL,
  MAXLOCKS     SMALLINT NOT NULL,
  LOCKS_AVAIL   INTEGER NOT NULL,
  CPU_SPEED     DOUBLE NOT NULL,
  REMARKS      VARCHAR(254),
  DBHEAP       INTEGER NOT NULL,
  COMM_SPEED    DOUBLE NOT
NULL,
  PARALLELISM   CHAR(2) NOT NULL,
  DATAJOINER   CHAR(1) NOT NULL,
  PRIMARY KEY
(EXPLAIN_REQUESTER,
  EXPLAIN_TIME,
  SOURCE_NAME,
  SOURCE_SCHEMA,
  SOURCE_VERSION))
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_STATEMENT
--
CREATE TABLE EXPLAIN_STATEMENT
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA   VARCHAR(128)
NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL  CHAR(1) NOT
NULL,
  STMTNO         INTEGER NOT NULL,
  SECTNO         INTEGER NOT NULL,
  QUERYNO        INTEGER NOT
NULL,
  QUERYTAG       CHAR(20) NOT
NULL,

```

```

STATEMENT_TYPE CHAR(2) NOT
NULL,
UPDATABLE      CHAR(1) NOT
NULL,
DELETABLE      CHAR(1) NOT NULL,
TOTAL_COST     DOUBLE NOT
NULL,
STATEMENT_TEXT CLOB(2M) NOT
NULL NOT LOGGED,
SNAPSHOT       BLOB(10M) NOT
LOGGED,
QUERY_DEGREE   INTEGER NOT
NULL,
PRIMARY KEY
(EXPLAIN_REQUESTER,
  EXPLAIN_TIME,
  SOURCE_NAME,
  SOURCE_SCHEMA,
  SOURCE_VERSION,
  STMTNO,
  SECTNO),
FOREIGN KEY
(EXPLAIN_REQUESTER,
  EXPLAIN_TIME,
  SOURCE_NAME,
  SOURCE_SCHEMA,
  SOURCE_VERSION)
REFERENCES EXPLAIN_INSTANCE
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_ARGUMENTS
--
CREATE TABLE EXPLAIN_ARGUMENT
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128)
NOT NULL,
  SOURCE_SCHEMA   VARCHAR(128)
NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL  CHAR(1) NOT
NULL,
  STMTNO         INTEGER NOT
NULL,
  SECTNO         INTEGER NOT
NULL,
  OPERATOR_ID    INTEGER NOT
NULL,
  ARGUMENT_TYPE  CHAR(8) NOT
NULL,
  ARGUMENT_VALUE VARCHAR(1024),
  LONG_ARGUMENT_VALUE CLOB(2M)
NOT LOGGED,
  FOREIGN KEY (EXPLAIN_REQUESTER,
  EXPLAIN_TIME,
  SOURCE_NAME,
  SOURCE_SCHEMA,

```

```

SOURCE_VERSION,
EXPLAIN_LEVEL,
STMTNO,
SECTNO)
REFERENCES EXPLAIN_STATEMENT
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_OBJECT
--
CREATE TABLE EXPLAIN_OBJECT ( EXPLAIN_REQUESTER
VARCHAR(128) NOT NULL,
  EXPLAIN_TIME    TIMESTAMP NOT
NULL,
  SOURCE_NAME     VARCHAR(128) NOT
NULL,
  SOURCE_SCHEMA   VARCHAR(128)
NOT NULL,
  SOURCE_VERSION  VARCHAR(64)
NOT NULL,
  EXPLAIN_LEVEL  CHAR(1) NOT
NULL,
  STMTNO         INTEGER NOT NULL,
  SECTNO         INTEGER NOT NULL,
  OBJECT_SCHEMA  VARCHAR(128)
NOT NULL,
  OBJECT_NAME    VARCHAR(128) NOT
NULL,
  OBJECT_TYPE    CHAR(2) NOT
NULL,
  CREATE_TIME    TIMESTAMP,
  STATISTICS_TIME TIMESTAMP,
  COLUMN_COUNT   SMALLINT NOT
NULL,
  ROW_COUNT      BIGINT NOT
NULL,
  WIDTH          INTEGER NOT NULL,
  PAGES          INTEGER NOT NULL,
  DISTINCT       CHAR(1) NOT NULL,
  TABLESPACE_NAME VARCHAR(128),
  OVERHEAD       DOUBLE NOT
NULL,
  TRANSFER_RATE  DOUBLE NOT
NULL,
  PREFETCHSIZE  INTEGER NOT
NULL,
  EXTENTSIZE     INTEGER NOT
NULL,
  CLUSTER        DOUBLE NOT NULL,
  NLEAF          INTEGER NOT NULL,
  NLEVELS        INTEGER NOT NULL,
  FULLKEYCARD    BIGINT NOT
NULL,
  OVERFLOW       INTEGER NOT
NULL,
  FIRSTKEYCARD   BIGINT NOT
NULL,
  FIRST2KEYCARD  BIGINT NOT
NULL,
  FIRST3KEYCARD  BIGINT NOT
NULL,

```

```

NULL,          FIRST4KEYCARD  BIGINT  NOT
NULL,          SEQUENTIAL_PAGES  INTEGER  NOT
NULL,          DENSITY          INTEGER  NOT NULL,
               STATS_SRC        CHAR(1)  NOT
NULL,          AVERAGE_SEQUENCE_GAP
DOUBLE NOT NULL, AVERAGE_SEQUENCE_FETCH_GAP
DOUBLE NOT NULL, AVERAGE_SEQUENCE_PAGES
DOUBLE NOT NULL, AVERAGE_SEQUENCE_FETCH_PAGES
DOUBLE NOT NULL, AVERAGE_RANDOM_PAGES
DOUBLE NOT NULL, AVERAGE_RANDOM_FETCH_PAGES
DOUBLE NOT NULL, NUMRIDS          BIGINT  NOT
NULL,          NUMRIDS_DELETED    BIGINT  NOT
NULL,          NUM_EMPTY_LEAFS    BIGINT
NOT NULL,      ACTIVE_BLOCKS      BIGINT  NOT
NULL,          FOREIGN KEY (EXPLAIN_REQUESTER,
                           EXPLAIN_TIME,
                           SOURCE_NAME,
                           SOURCE_SCHEMA,
                           SOURCE_VERSION,
                           EXPLAIN_LEVEL,
                           STMTNO,
                           SECTNO)
               REFERENCES EXPLAIN_STATEMENT
               ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_OPERATOR
--
CREATE TABLE EXPLAIN_OPERATOR
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME      TIMESTAMP NOT
NULL,          SOURCE_NAME    VARCHAR(128) NOT
NULL,          SOURCE_SCHEMA  VARCHAR(128)
NOT NULL,      SOURCE_VERSION VARCHAR(64)
NOT NULL,      EXPLAIN_LEVEL CHAR(1) NOT
NULL,          STMTNO        INTEGER  NOT NULL,
               SECTNO        INTEGER  NOT NULL,
               OPERATOR_ID   INTEGER  NOT
NULL,          OPERATOR_TYPE  CHAR(6)  NOT
NULL,          TOTAL_COST     DOUBLE  NOT
NULL,

```

```

IO_COST        DOUBLE  NOT NULL,
CPU_COST       DOUBLE  NOT NULL,
FIRST_ROW_COST DOUBLE  NOT
NULL,          RE_TOTAL_COST  DOUBLE  NOT
NULL,          RE_IO_COST    DOUBLE  NOT
NULL,          RE_CPU_COST   DOUBLE  NOT
NULL,          COMM_COST     DOUBLE  NOT
NULL,          FIRST_COMM_COST DOUBLE  NOT
NULL,          BUFFERS       DOUBLE  NOT NULL,
               REMOTE_TOTAL_COST DOUBLE
NOT NULL,      REMOTE_COMM_COST DOUBLE
NOT NULL,      FOREIGN KEY (EXPLAIN_REQUESTER,
                           EXPLAIN_TIME,
                           SOURCE_NAME,
                           SOURCE_SCHEMA,
                           SOURCE_VERSION,
                           EXPLAIN_LEVEL,
                           STMTNO,
                           SECTNO)
               REFERENCES EXPLAIN_STATEMENT
               ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_PREDICATE
--
CREATE TABLE EXPLAIN_PREDICATE
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME      TIMESTAMP NOT
NULL,          SOURCE_NAME    VARCHAR(128) NOT
NULL,          SOURCE_SCHEMA  VARCHAR(128)
NOT NULL,      SOURCE_VERSION VARCHAR(64)
NOT NULL,      EXPLAIN_LEVEL CHAR(1) NOT
NULL,          STMTNO        INTEGER  NOT NULL,
               SECTNO        INTEGER  NOT NULL,
               OPERATOR_ID   INTEGER  NOT
NULL,          PREDICATE_ID  INTEGER  NOT
NULL,          HOW_APPLIED   CHAR(5)  NOT
NULL,          WHEN_EVALUATED CHAR(3)  NOT
NULL,          RELOP_TYPE    CHAR(2)  NOT
NULL,          SUBQUERY      CHAR(1)  NOT
NULL,          FILTER_FACTOR  DOUBLE  NOT
NULL,

```

```

LOGGED,        PREDICATE_TEXT CLOB(2M) NOT
FOREIGN KEY (EXPLAIN_REQUESTER,
             EXPLAIN_TIME,
             SOURCE_NAME,
             SOURCE_SCHEMA,
             SOURCE_VERSION,
             EXPLAIN_LEVEL,
             STMTNO,
             SECTNO)
             REFERENCES EXPLAIN_STATEMENT
             ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- EXPLAIN_STREAM
--
CREATE TABLE EXPLAIN_STREAM
( EXPLAIN_REQUESTER VARCHAR(128) NOT NULL,
  EXPLAIN_TIME      TIMESTAMP NOT
NULL,          SOURCE_NAME    VARCHAR(128) NOT
NULL,          SOURCE_SCHEMA  VARCHAR(128) NOT
NULL,          SOURCE_VERSION VARCHAR(64) NOT
NULL,          EXPLAIN_LEVEL CHAR(1) NOT
NULL,          STMTNO        INTEGER  NOT NULL,
               SECTNO        INTEGER  NOT NULL,
               STREAM_ID     INTEGER  NOT NULL,
               SOURCE_TYPE   CHAR(1)  NOT
NULL,          SOURCE_ID     INTEGER  NOT NULL,
               TARGET_TYPE   CHAR(1)  NOT NULL,
               TARGET_ID     INTEGER  NOT NULL,
               OBJECT_SCHEMA VARCHAR(128),
               OBJECT_NAME   VARCHAR(128),
               STREAM_COUNT  DOUBLE  NOT
NULL,          COLUMN_COUNT  SMALLINT NOT
NULL,          PREDICATE_ID  INTEGER  NOT
NULL,          COLUMN_NAMES  CLOB(2M) NOT
LOGGED,        PMID          SMALLINT NOT NULL,
               SINGLE_NODE  CHAR(5),
               PARTITION_COLUMNS CLOB(2M) NOT
LOGGED,        FOREIGN KEY (EXPLAIN_REQUESTER,
                           EXPLAIN_TIME,
                           SOURCE_NAME,
                           SOURCE_SCHEMA,
                           SOURCE_VERSION,
                           EXPLAIN_LEVEL,
                           STMTNO,
                           SECTNO)
               REFERENCES EXPLAIN_STATEMENT
               ON DELETE CASCADE)
IN USERSPACE1

```

```

INDEX IN USERSPACE1;
--
-- ADVISE TABLES
--
-- ADVISE_INSTANCE
-- (must be defined first due to referential integrity definitions)
--
CREATE TABLE ADVISE_INSTANCE (
  START_TIME    TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
  END_TIME      TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
  MODE          VARCHAR(4) NOT NULL WITH
DEFAULT "",
  WKLD_COMPRESSION CHAR(4) NOT NULL WITH
DEFAULT 'NONE',
  STATUS        CHAR(9) NOT NULL WITH DEFAULT
",
  PRIMARY KEY (START_TIME))
IN USERSPACE1
INDEX IN USERSPACE1;

--
-- ADVISE_INDEX
--
CREATE TABLE ADVISE_INDEX(
  EXPLAIN_REQUESTER VARCHAR(128) NOT NULL
WITH DEFAULT "",
  EXPLAIN_TIME        TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
  SOURCE_NAME         VARCHAR(128) NOT NULL WITH
DEFAULT "",
  SOURCE_SCHEMA       VARCHAR(128) NOT NULL WITH
DEFAULT "",
  SOURCE_VERSION      VARCHAR(64) NOT NULL WITH
DEFAULT "",
  EXPLAIN_LEVEL       CHAR(1) NOT NULL WITH
DEFAULT "",
  STMTNO              INTEGER NOT NULL WITH
DEFAULT 0,
  SECTNO              INTEGER NOT NULL WITH
DEFAULT 0,
  QUERYNO             INTEGER NOT NULL WITH
DEFAULT 0,
  QUERYTAG            CHAR(20) NOT NULL WITH
DEFAULT "",
  NAME                VARCHAR(128) NOT NULL,
  CREATOR             VARCHAR(128) NOT NULL WITH
DEFAULT "",
  TBNAME              VARCHAR(128) NOT NULL,
  TBcreator           VARCHAR(128) NOT NULL WITH
DEFAULT "",
  COLNAMES            CLOB(2M) NOT NULL,
  UNIQUERULE          CHAR(1) NOT NULL WITH
DEFAULT "",
  COLCOUNT           SMALLINT NOT NULL WITH
DEFAULT 0,
  IID                 SMALLINT NOT NULL WITH DEFAULT 0,
  NLEAF               INTEGER NOT NULL WITH DEFAULT
0,

```

```

  NLEVELS            SMALLINT NOT NULL WITH
DEFAULT 0,
  FIRSTKEYCARD       BIGINT NOT NULL WITH
DEFAULT 0,
  FULLKEYCARD        BIGINT NOT NULL WITH
DEFAULT 0,
  CLUSTERRATIO       SMALLINT NOT NULL WITH
DEFAULT 0,
  CLUSTERFACTOR      DOUBLE NOT NULL WITH
DEFAULT 0,
  USERDEFINED        SMALLINT NOT NULL WITH
DEFAULT 0,
  SYSTEM_REQUIRED    SMALLINT NOT NULL WITH
DEFAULT 0,
  CREATE_TIME        TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
  STATS_TIME         TIMESTAMP WITH DEFAULT
CURRENT TIMESTAMP,
  PAGE_FETCH_PAIRS   VARCHAR(254) NOT NULL
WITH DEFAULT "",
  REMARKS            VARCHAR(254) WITH DEFAULT
",
  DEFINER            VARCHAR(128) NOT NULL WITH
DEFAULT "",
  CONVERTED          CHAR(1) NOT NULL WITH
DEFAULT "",
  SEQUENTIAL_PAGES   INTEGER NOT NULL WITH
DEFAULT 0,
  DENSITY            INTEGER NOT NULL WITH
DEFAULT 0,
  FIRST2KEYCARD      BIGINT NOT NULL WITH
DEFAULT 0,
  FIRST3KEYCARD      BIGINT NOT NULL WITH
DEFAULT 0,
  FIRST4KEYCARD      BIGINT NOT NULL WITH
DEFAULT 0,
  PCTFREE            SMALLINT NOT NULL WITH
DEFAULT -1,
  UNIQUE_COLCOUNT   SMALLINT NOT NULL WITH
DEFAULT -1,
  MINPCTUSED         SMALLINT NOT NULL WITH
DEFAULT 0,
  REVERSE_SCANS      CHAR(1) NOT NULL WITH
DEFAULT 'N',
  USE_INDEX          CHAR(1),
  CREATION_TEXT      CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",
  PACKED_DESC        BLOB(1M) NOT LOGGED,
  RUN_ID             TIMESTAMP,
  INDEXTYPE          VARCHAR(4) NOT NULL WITH
DEFAULT "",
  EXISTS             CHAR(1) NOT NULL WITH DEFAULT
'N',
  RIDTOBLOCK         CHAR(1) NOT NULL WITH
DEFAULT 'N',
  FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--

```

```

-- ADVISE_WORKLOAD
--
CREATE TABLE ADVISE_WORKLOAD (
  WORKLOAD_NAME     CHAR(128) NOT NULL WITH
DEFAULT 'WK0',
  STATEMENT_NO      INTEGER NOT NULL WITH
DEFAULT 1,
  STATEMENT_TEXT    CLOB(2M) NOT NULL NOT
LOGGED,
  STATEMENT_TAG     VARCHAR(256) NOT NULL WITH
DEFAULT "",
  FREQUENCY         INTEGER NOT NULL WITH
DEFAULT 1,
  IMPORTANCE        DOUBLE NOT NULL WITH
DEFAULT 1,
  WEIGHT            DOUBLE NOT NULL WITH DEFAULT
1,
  COST_BEFORE       DOUBLE,
  COST_AFTER        DOUBLE,
  COMPILABLE        CHAR(17))
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_MQT
--
CREATE TABLE ADVISE_MQT (
  EXPLAIN_REQUESTER VARCHAR(128) NOT NULL
WITH DEFAULT "",
  EXPLAIN_TIME        TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
  SOURCE_NAME         VARCHAR(128) NOT NULL WITH
DEFAULT "",
  SOURCE_SCHEMA       VARCHAR(128) NOT NULL WITH
DEFAULT "",
  SOURCE_VERSION      VARCHAR(64) NOT NULL WITH
DEFAULT "",
  EXPLAIN_LEVEL       CHAR(1) NOT NULL WITH
DEFAULT "",
  STMTNO              INTEGER NOT NULL WITH
DEFAULT 0,
  SECTNO              INTEGER NOT NULL WITH
DEFAULT 0,
  NAME                VARCHAR(128) NOT NULL,
  CREATOR             VARCHAR(128) NOT NULL WITH
DEFAULT "",
  IID                 SMALLINT NOT NULL WITH DEFAULT 0,
  CREATE_TIME        TIMESTAMP NOT NULL WITH
DEFAULT CURRENT TIMESTAMP,
  STATS_TIME         TIMESTAMP WITH DEFAULT
CURRENT TIMESTAMP,
  NUMROWS            DOUBLE NOT NULL WITH
DEFAULT 0,
  NUMCOLS            SMALLINT NOT NULL WITH
DEFAULT 0,
  ROWSIZE            DOUBLE NOT NULL WITH
DEFAULT 0,
  BENEFIT            FLOAT NOT NULL WITH DEFAULT
0.0,
  USE_MQT            CHAR(1),
  MQT_SOURCE         CHAR(1),
  QUERY_TEXT         CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT "",

```

```

CREATION_TEXT CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
SAMPLE_TEXT CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
COLSTATS CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
EXTRA_INFO BLOB(2M) NOT NULL NOT
LOGGED WITH default BLOB(""),
TBSPACE VARCHAR(128) NOT NULL WITH
DEFAULT ",
RUN_ID TIMESTAMP,
REFRESH_TYPE CHAR(1) NOT NULL WITH
DEFAULT ",
EXISTS CHAR(1) NOT NULL WITH DEFAULT
'N',
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_PARTITION
--
CREATE TABLE ADVISE_PARTITION (
EXPLAIN_REQUESTER VARCHAR(128) NOT NULL
WITH DEFAULT ",
EXPLAIN_TIME TIMESTAMP NOT NULL WITH
DEFAULT CURRENT_TIMESTAMP,
SOURCE_NAME VARCHAR(128) NOT NULL WITH
DEFAULT ",
SOURCE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT ",
SOURCE_VERSION VARCHAR(64) NOT NULL WITH
DEFAULT ",
EXPLAIN_LEVEL CHAR(1) NOT NULL WITH
DEFAULT ",
STMTNO INTEGER NOT NULL WITH
DEFAULT 0,
SECTNO INTEGER NOT NULL WITH
DEFAULT 0,
QUERYNO INTEGER NOT NULL WITH
DEFAULT 0,
QUERYTAG CHAR(20) NOT NULL WITH
DEFAULT ",
TBNAME VARCHAR(128) NOT NULL,
TBCREATOR VARCHAR(128) NOT NULL WITH
DEFAULT ",
PMID SMALLINT NOT NULL,
TBSPACE VARCHAR(128) NOT NULL WITH
DEFAULT ",
COLNAMES CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
COLCOUNT SMALLINT NOT NULL WITH
DEFAULT 0,
REPLICATE CHAR(1) NOT NULL WITH
DEFAULT 'N',
COST DOUBLE NOT NULL,
USEIT CHAR(1),
RUN_ID TIMESTAMP,
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- ADVISE_TABLE
--
CREATE TABLE ADVISE_TABLE (
RUN_ID TIMESTAMP,
TABLE_NAME VARCHAR(128) NOT NULL,
TABLE_SCHEMA VARCHAR(128) NOT NULL WITH
DEFAULT ",
TABLESPACE VARCHAR(128) NOT NULL WITH
DEFAULT ",
SELECTION_FLAG VARCHAR(8) NOT NULL WITH
DEFAULT ",
TABLE_EXISTS CHAR(1) NOT NULL WITH
DEFAULT ",
USE_TABLE CHAR(1) NOT NULL WITH
DEFAULT ",
GEN_COLUMNS CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
ORGANIZE_BY CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
CREATION_TEXT CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
ALTER_COMMAND CLOB(2M) NOT NULL NOT
LOGGED WITH DEFAULT ",
DISKUSE DOUBLE NOT NULL WITH
DEFAULT 0,
FOREIGN KEY (RUN_ID)
REFERENCES ADVISE_INSTANCE
(START_TIME)
ON DELETE CASCADE)
IN USERSPACE1
INDEX IN USERSPACE1;
--
-- Commit work
--
COMMIT WORK;
--
-- Optional Indexes: The following indexes are recommended
for improved performance
-- of explain-related utilities. These create index statements can
be deleted, or
-- the indexes dropped if space is a problem.
--
CREATE INDEX STMT_I1 on
EXPLAIN_STATEMENT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX ARG_I1 on
EXPLAIN_ARGUMENT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX PRD_I1 on
EXPLAIN_PREDICATE(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX OPR_I1 on
EXPLAIN_OPERATOR(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO, OPERATOR_ID);
CREATE INDEX STM_I1 on
EXPLAIN_STREAM(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX OBJ_I1 on
EXPLAIN_OBJECT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX IDX_I1 on
ADVISE_INDEX (EXPLAIN_TIME);
CREATE INDEX IDX_I2 on
ADVISE_INDEX (NAME, EXPLAIN_TIME);
CREATE INDEX MQT_I1 on
ADVISE_MQT (EXPLAIN_TIME);
CREATE INDEX MQT_I2 on
ADVISE_MQT (NAME, EXPLAIN_TIME);
CREATE INDEX PRT_I1 on
ADVISE_PARTITION (EXPLAIN_TIME);
--
-- Commit work
--
COMMIT WORK;

```

```

EXPLAIN_STREAM(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX OBJ_I1 on
EXPLAIN_OBJECT(EXPLAIN_TIME, EXPLAIN_LEVEL,
STMTNO, SECTNO);
CREATE INDEX IDX_I1 on
ADVISE_INDEX (EXPLAIN_TIME);
CREATE INDEX IDX_I2 on
ADVISE_INDEX (NAME, EXPLAIN_TIME);
CREATE INDEX MQT_I1 on
ADVISE_MQT (EXPLAIN_TIME);
CREATE INDEX MQT_I2 on
ADVISE_MQT (NAME, EXPLAIN_TIME);
CREATE INDEX PRT_I1 on
ADVISE_PARTITION (EXPLAIN_TIME);
--
-- Commit work
--
COMMIT WORK;

```

utils/UNEXPLAIN.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
-- 2004
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.

```

```

DROP INDEX STMT_I1;
DROP INDEX ARG_I1;
DROP INDEX PRD_I1;
DROP INDEX OPR_I1;
DROP INDEX STM_I1;
DROP INDEX OBJ_I1;
DROP TABLE EXPLAIN_INSTANCE;
DROP TABLE EXPLAIN_STATEMENT;
DROP TABLE EXPLAIN_ARGUMENT;
DROP TABLE EXPLAIN_OBJECT;
DROP TABLE EXPLAIN_OPERATOR;
DROP TABLE EXPLAIN_PREDICATE;
DROP TABLE EXPLAIN_STREAM;
DROP TABLE ADVISE_INDEX;
DROP TABLE ADVISE_WORKLOAD;

```

tpccCom/comreg.h

```

// compreg.h : Declaration of the CCompReg
#pragma once
#include "resource.h" // main symbols
#include "tpccCom.h"

// CCompReg

```

TPC Benchmark™ C Full Disclosure Report - IBM eServer p5 595 Model 9119-595

```

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 )
                            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/dlldatax.h

```

#pragma once
#ifdef _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
//
#ifdef 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/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
#ifdef 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.
#ifdef 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
#ifdef _WIN32_WINNT          // Allow use of features
specific to Windows NT 4 or later.
#define _WIN32_WINNT 0x0400 // Change this to the
appropriate value to target Windows 2000 or later.
#endif
#ifdef _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
#ifdef _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_NAMESPACES
#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/tpccCom.h

```

// tpcc_com.h : Declaration of the Ctpcc_com
#pragma once
#include "tpccCom.h"
#include "resource.h" // main symbols
#include <comsvcs.h>
#include "..\tpccclsapi\tpcc.h"
#include <db2tpcc.h>
#include <tpcc.h>
#define NULL_DB "nullDB"
static HINSTANCE dbInstance = NULL;
static CRITICAL_SECTION debugMutex;
static CRITICAL_SECTION errorMutex;
static int comServerID = 0;
static ofstream debugStream;
static ofstream errorStream;
static int debugFileOpen = 0;
static int errorFileOpen = 0;
static int nullDB = 0;
static char dbName[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);
NORD_PTR do_nord;
PYMT_PTR do_pymt;
ORDS_PTR do_ords;
STOK_PTR do_stok;
CONNECT_PTR do_connect;
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 Itpc_com
{
public:
Ctpcc_com()
{
int rc = ERR;
connected = 0;
connectHandleInUse = 0;
if(debugFlag)
{
if(!debugFileOpen)
{
InitializeCriticalSection(&debugMutex);
//open comLog
char comLogFile[128];

sprintf(comLogFile, "C:\\inetpub\\wwwroot\\tpcc\\comLog_d
ebug.txt");
debugStream.rdbuf( )-
>open(comLogFile, ios_base::in | ios_base::out | ios_base::app);
debugFileOpen = 1;
}
}
//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 |
ios_base::app);
errorFileOpen=1;
}
//get registry values

```

```

if((rc = readRegistry()) != OK)
{
ERRORMSG("Unable to open registry key "
<< REGISTRY_SUB_KEY << " rc:" << rc << endl);
return;
}

DEBUGMSG("nullDB:" <<nullDB<<"
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
EnterCriticalSection(&errorMutex);
if((rc = connectDB()) != OK)
{
ERRORMSG("unable to connect to db
"<<dbName<<" rc : "<<rc <<endl);
LeaveCriticalSection(&errorMutex);
return;
}
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(Itpc_com)
COM_INTERFACE_ENTRY(IObjectControl)
END_COM_MAP()
// IObjectControl
public:
STDMETHOD(Activate)();
STDMETHOD_(BOOL, CanBePooled)();
STDMETHOD_(void, Deactivate)();
CComPtr<IObjectContext> m_spObjectContext;

// Itpc_com
public:
STDMETHOD(doStockLevel)(INT *size, UCHAR **buffer);
STDMETHOD(doNewOrder)(INT* size, UCHAR** buffer);
STDMETHOD(doPayment)(INT* size, UCHAR** buffer);

```

```

        STDMETHOD(doOrderStatus)(INT* size, UCHAR**
buffer);
        STDMETHOD(doDBInfo)(void);
        STDMETHOD(doSetComplete)(void);
        int connected;
        int connectHandleInUse;
private:
        //db2 specific context
        void *connectHandle;
        int loadLibrary();
        int readRegistry();
        int connectDB();
};
OBJECT_ENTRY_AUTO(__uuidof(tpcc_com), Ctpcc_com)

```

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 "..\tpcc\sapi\tpcc.h"
#include <db2tpcc.h>
#include <tpcc.h>
#define NULL_DB "nullDB"
static HINSTANCE dbInstance = NULL;
static CRITICAL_SECTION debugMutex;
static CRITICAL_SECTION errorMutex;
static int comServerID = 0;
static ofstream debugStream;
static ofstream errorStream;
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)(pymt_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);
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 tpcc_com
{
public:
        Ctpcc_com()
        {
                int rc = ERR;
                connected = 0;
                connectHandleInUse = 0;
                if(debugFlag)
                {
                        if(!debugFileOpen)
                        {
                                InitializeCriticalSection(&debugMutex);
                                //open comLog
                                char comLogFile[128];

                                sprintf(comLogFile,"C:\Inetpub\wwwroot\tpcc\comLog_d
ebug.txt");
                                debugStream.rdbuf( )
>open(comLogFile,ios_base::in | ios_base::out | ios_base::app);
                                debugFileOpen = 1;
                        }
                }

                //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 |
ios_base::app);
                                errorFileOpen=1;
                }

                //get registry values
                if((rc = readRegistry()) != OK)
                {
                        ERRORMSG("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
        EnterCriticalSection(&errorMutex);
        if((rc = connectDB()) != OK)

```

```

        {
                ERRORMSG("unable to connect to db
"<<dbName<<" rc :"<<rc <<endl);
                LeaveCriticalSection(&errorMutex);
                return;
        }
        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:
                STDMETHOD(Activate)();
                STDMETHOD_(BOOL, CanBePooled)();
                STDMETHOD_(void, Deactivate)();
                CComPtr<IObjectContext> m_spObjectContext;

// Itpcc_com
        public:
                STDMETHOD(doStockLevel)(INT *size, UCHAR **buffer);
                STDMETHOD(doNewOrder)(INT* size, UCHAR** buffer);
                STDMETHOD(doPayment)(INT* size, UCHAR** buffer);
                STDMETHOD(doOrderStatus)(INT* size, UCHAR**
buffer);
                STDMETHOD(doDBInfo)(void);
                STDMETHOD(doSetComplete)(void);
                int connected;
                int connectHandleInUse;
        private:
                //db2 specific context
                void *connectHandle;
                int loadLibrary();
                int readRegistry();
                int connectDB();

};
OBJECT_ENTRY_AUTO(__uuidof(tpcc_com), Ctpcc_com)

```

tpccCom/tpccCom.def

```

; tpccCom.def : Declares the module parameters.
LIBRARY "tpccCom.DLL"
EXPORTS
        DllCanUnloadNow PRIVATE
        DllGetClassObject PRIVATE

```

```

DllRegisterServer PRIVATE
DllUnregisterServer PRIVATE

```

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 "oidl.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)* pbstrCLSIDs, [out] SAFEARRAY(BSTR)*
pbstrDescriptions);
    [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/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/comreg.cpp

```

// compreg.cpp : Implementation of CCompReg
#include "stdafx.h"
#include "comreg.h"

```

```

// CCompReg

```

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/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 "comreg.h"
#include "dlldata.h"
class CtpccComModule : public CAtdllModuleT<
CtpccComModule >
{
public :
    DECLARE_LIBID(LIBID_tpccComLib)
    DECLARE_REGISTRY_APPID_RESOURCEID(IDR_TPC
CCOM, "{11ED2355-1A27-42F1-ADFF-F201F5E82BCE}")
};
CtpccComModule _AtlModule;

// 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;
    return _AtlModule.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 _AtlModule.DllCanUnloadNow();
}

```



```
// Returns a class factory to create an object of the requested
type
STDAPI DllGetObject(REFCLSID rclsid, REFIID riid,
LPVOID* ppv)
{
#ifdef _MERGE_PROXYSTUB
    if (PrxDllGetObject(rclsid, riid, ppv) == S_OK)
        return S_OK;
#endif
    return _AtlModule.DllGetObject(rclsid, riid, ppv);
}
```

```
// DllRegisterServer - Adds entries to the system registry
STDAPI DllRegisterServer(void)
{
    // registers object, typelib and all interfaces in typelib
    HRESULT hr = _AtlModule.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 = _AtlModule.DllUnregisterServer();
#ifdef _MERGE_PROXYSTUB
    if (FAILED(hr))
        return hr;
    hr = PrxDllRegisterServer();
    if (FAILED(hr))
        return hr;
    hr = PrxDllUnregisterServer();
#endif
    return hr;
}
```

tpccCom/tpcc_com.cpp

```
// tpcc_com.cpp : Implementation of Ctpcc_com
#include "stdafx.h"
#include "tpcc_com.h"
#include ".\tpcc_com.h"
#include <db2tpcc.h>
// Ctpcc_com
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 :
**           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::doStockLevel(INT *size, UCHAR
**buffer)
{
    stok_wrapper * stok;
    stok = (stok_wrapper *) *buffer;

    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);

    do_stok(stok,connectHandle);
    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;
    return S_OK;
}
/*
*****
** Name      : doNewOrder
** 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::doNewOrder(INT* size,
UCHAR** buffer)
{
    nord_wrapper *nord;
    nord = (nord_wrapper *) *buffer;
    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);

    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);
    DEBUGMSG("Connection handle set to free" << endl);
    connectHandleInUse = 0;
    return S_OK;
}
/*
*****
** Name      :    doPayment
** 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::doPayment(INT* size, UCHAR**
buffer)
{
    paym_wrapper *pymt;
    pymt = (paym_wrapper *) *buffer;
    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;

    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;
    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);
}

```

```

do_ords(ords, connectHandle);
    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;

    return S_OK;
}
/*
*****
** Name      :    doDBInfo
** Description :    Function to test com interface
** Parameters :
** Returns    :    int - return code
** Comments   :
**
*****
*/
STDMETHODIMP Ctpcc_com::doDBInfo(void)
{
    DEBUGMSG("Stub function to warm object pool" << endl);
    return S_OK;
}
/*
*****
** Name      :    loadLibrary
** Description :    Function loads appropiate db library
based on
**            registry setting
** Parameters :
** Returns    :    int - return code
** Comments   :
**
*****
*/
Ctpcc_com::loadLibrary()
{
    DEBUGMSG("Entered loadLibrary function" << endl);
    //check to see if dbInstance is already loaded
    if(!dbInstance)
    {
        DEBUGMSG("Database dll not loaded. Loading
dll." << endl);
        if (nullDB)
        {
            DEBUGMSG("Loading "<< dbType << " nulldb
dll." << endl);
            dbInstance =
LoadLibrary("c:\\inetpub\\wwwroot\\tpcc\\nullDB.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;
    }
    DEBUGMSG(dbType << " nulldb dll
loaded"<<endl);
    }
    else if(strcmp(dbType,"DB2") == 0)
    {
        DEBUGMSG("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:" << GetLastError() << endl);

            return ERR_DB2_DLL_NOT_LOADED;
        }
        DEBUGMSG(dbType<< " dll loaded"<<endl);
    }
    else if( strcmp(dbType,"ORACLE") == 0 )
    {
        DEBUGMSG("Unable to load oracle
dll:"<<dbType<<endl);
        ERRORMSG("Unable to load oracle dll,
rc:"<<GetLastError())<<endl);
        return ERR_ORACLE_DLL_NOT_LOADED;
    }
    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);
    DEBUGMSG("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("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_orcls function address
from "<<dbType<<" dll"<<endl);
    if( do_orcls = (ORDS_PTR)
GetProcAddress(dbInstance,"do_orcls") == NULL)
        return
ERR_ORCLS_ADDRESS_NOT_FOUND;
    DEBUGMSG("do_orcls function
address:"<<DEBUGADDRESS(do_orcls)<<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);
    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
*****
*/
Ctpcc_com::readRegistry()
{
    //open registry key
    HKEY    registryKey;
    DWORD   regType;
    char    value[MAX_STRING_LEN];
    DWORD   regValue;
    DWORD   regValueSize = MAX_STRING_LEN;
    DEBUGMSG("Entered readRegistry(), opening key:"<<
REGISTRY_SUB_KEY <<endl);

```

```

//open up registry key
if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTER
Y_SUB_KEY,0,KEY_READ,&registryKey) ==
ERROR_SUCCESS)
{
    DEBUGMSG(REGISTRY_SUB_KEY<<" open,
getting database type from key"<<endl);
    regValueSize = sizeof(value);
    if
(RegQueryValueEx(registryKey,DB_TYPE,0,&regType,(BYTE
*)&value,&regValueSize)== ERROR_SUCCESS )
        strcpy(dbType,value);
    DEBUGMSG("Database type:"<<dbType<<" from
registry key."<<endl);
    DEBUGMSG("Getting database name from registry
key."<<endl);
    regValueSize = sizeof(value);
    if
(RegQueryValueEx(registryKey,DB_NAME,0,&regType,(BYTE
*)&value,&regValueSize)== ERROR_SUCCESS )
        strcpy(dbName,value);
    DEBUGMSG("Database name:"<<dbName<<endl);
    DEBUGMSG("Getting null database flag from
key."<<endl);
    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NULL_DB,0,&regType,(
BYTE *)&regValue,&regValueSize) == ERROR_SUCCESS)
        nullDB = regValue;
    DEBUGMSG("Null database flag:"<<nullDB<<endl);
    return OK;
}
DEBUGMSG("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  :
*****
*/
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)
        {

```

```

        DEBUGMSG("Setting Context handle in use to
true"<<endl);
        connectHandleInUse = 1;
        connected =
do_connection(dbName,&connectHandle);
        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/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/dlldatax.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_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 6.00.0361 */
/* at Wed Feb 11 08:32:46 2004
*/
/* 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
*/

#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
#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_
#ifndef __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,0x4
4,0x00,0xaa,0x00,0xb6,0x77,0x0a);

MIDL_DEFINE_GUID(IID,
IID_Itpcc_com,0x5B4FA473,0x2E68,0x4D79,0xA6,0x26,0xF3,0
x8B,0x30,0xB8,0x19,0x6E);

MIDL_DEFINE_GUID(IID,
LIBID_tpccComLib,0x91F1B8B0,0x89E9,0x457B,0xA2,0x28,0x
3E,0x2D,0x6C,0xE3,0xE7,0x52);

MIDL_DEFINE_GUID(CLSID,
CLSID_CompReg,0x90EEDAFF,0xF8D3,0x4711,0x99,0xA9,0x
8A,0xC3,0xC0,0xFE,0x5D,0xB9);

MIDL_DEFINE_GUID(CLSID,
CLSID_tpcc_com,0x5F752BF2,0xF739,0x43D4,0x84,0x92,0x4
4,0xC1,0x95,0x81,0xC0,0xA1);
#undef MIDL_DEFINE_GUID
#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/

```

tpccCom/tpccCom_p.c

```

/* this ALWAYS GENERATED file contains the proxy stub code
*/

/* File created by MIDL compiler version 6.00.0361 */
/* at Wed Feb 11 08:32:46 2004
*/
/* 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: 4100 ) /* unreferenced arguments in
x86 call */
#pragma warning( disable: 4211 ) /* redefine extent to static */
#pragma warning( disable: 4232 ) /* dllimport identity*/
#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 TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 2
typedef struct _MIDL_TYPE_FORMAT_STRING
{
short Pad;
unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;
typedef struct _MIDL_PROC_FORMAT_STRING
{
short Pad;
unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x
10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

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 Itppc_com_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
Itppc_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 die there with the
RPC_X_WRONG_STUB_VERSION error.
#endif

static const MIDL_PROC_FORMAT_STRING
__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, */
/* 16 */ 0x2, /* 2 */
/* 18 */ 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 */
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 GetComponents */
/* 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( 0x24 ), /* 36 */
/* 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 bstrDescriptions */
/* 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( 0x1 ), /* 1 */

```

```

/* 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, */
0x3, /* 3 */
/* 226 */ 0x8, /* 8 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 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 */
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 */
0x0, /* 0 */
/* Procedure doNewOrder */
/* 252 */ 0x33, /* FC_AUTO_HANDLE */
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, */
0x3, /* 3 */
/* 268 */ 0x8, /* 8 */

```

```

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 */
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 */
0x0, /* 0 */
/* Procedure doPayment */
/* 294 */ 0x33, /* FC_AUTO_HANDLE */
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, */
0x3, /* 3 */
/* 310 */ 0x8, /* 8 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 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 */
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 */
0x0, /* 0 */
/* Procedure doOrderStatus */
/* 336 */ 0x33, /* FC_AUTO_HANDLE */
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, */
0x3, /* 3 */
/* 352 */ 0x8, /* 8 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 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 */
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 */
0x0, /* 0 */
/* Procedure doDBInfo */
/* 378 */ 0x33, /* FC_AUTO_HANDLE */
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, */
0x1, /* 1 */
/* 394 */ 0x8, /* 8 */
0x1, /* Ext Flags: new corr desc, */
/* 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 */
0x0, /* 0 */
0x0
}
};
static const MIDL_TYPE_FORMAT_STRING
__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( 0xffc ), /* -4 */
/* 14 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 16 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 18 */
0x17, /* FC_CSTRUCT */
0x3, /* 3 */
/* 20 */ NdrFcShort( 0x8 ), /* 8 */
/* 22 */ NdrFcShort( 0xffff ), /* 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( 0xff82 ), /* Offset= -126 (18) */
/* 146 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 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( 0xffd2 ), /* Offset= -46 (118) */
/* 166 */
0x5b, /* FC_END */
0x8, /* FC_LONG */
/* 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( 0xffffffff ), /* -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( 0xffffffff ), /* -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( 0xff8 ), /* -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( 0xff3c ), /* Offset= -196 (170) */
/* 368 */ NdrFcLong( 0x9 ), /* 9 */
/* 372 */ NdrFcShort( 0xff6e ), /* 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, /*
/* 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( 0xfe30 ), /* 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( 0xfffd4 ), /* 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( 0xfe30 ), /* Offset= -464 (282) */
/* 748 */ 0x5c, /* FC_PAD */
    0x5b, /* FC_END */
    0x1b, /* FC_CARRAY */
    0x3, /* 3 */
/* 752 */ NdrFcShort( 0x4 ), /* 4 */
/* 754 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/*
    0x0, /*
/* 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 */
    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 */
/* 794 */ 0x5b, /* FC_END */

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 */
0x8, /* FC_LONG */
/* 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_RP */
/* 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( 0xffff1 ), /* 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_RP */
/* 882 */ NdrFcShort( 0xfd4a ), /* Offset= -694 (188) */
/* 884 */

0x1b, /* FC_CARRAY */
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 */
0x8, /* FC_LONG */
/* 914 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */

/* 916 */

0x1b, /* FC_CARRAY */
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 */
0x8, /* FC_LONG */
/* 946 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */

/* 948 */

0x1b, /* FC_CARRAY */
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 */
0x8, /* FC_LONG */
/* 978 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */

/* 980 */

0x1b, /* FC_CARRAY */
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 */

```

```

/* 998 */      0x5c, /* FC_PAD */
              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 */
              0x8, /* FC_LONG */
/* 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_CARRAY */
              0x3, /* 3 */
/* 1022 */ NdrFcShort( 0x8 ), /* 8 */
/* 1024 */ 0x7, /* Corr desc: FC_USHORT */
              0x0, /* 0 */
/* 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,
    __MIDL_ProcFormatString.Format,
    &IComponentRegistrar_FormatStringOffsetTable[-3],
    0,
    0,
    0
};
static const MIDL_SERVER_INFO
IComponentRegistrar_ServerInfo =
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &IComponentRegistrar_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0;
};
INTERFACE_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,0x
30,0xB8,0x19,0x6E}} */
#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,
    __MIDL_ProcFormatString.Format,
    &Itpcc_com_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

static const MIDL_SERVER_INFO Itpcc_com_ServerInfo =
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &Itpcc_com_FormatStringOffsetTable[-3],
    0,
    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,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x6000169, /* MIDL Version 6.0.361 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};
const CInterfaceProxyVtbl * _tpccCom_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_Itpcc_comProxyVtbl,
    ( CInterfaceProxyVtbl *) &_IComponentRegistrarProxyVtbl,
    0
};
const CInterfaceStubVtbl * _tpccCom_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_Itpcc_comStubVtbl,
    ( CInterfaceStubVtbl *) &_IComponentRegistrarStubVtbl,
    0
};
PCInterfaceName const _tpccCom_InterfaceNamesList[] =
{
    "Itpcc_com",
    "IComponentRegistrar",
    0
};
const IID * _tpccCom_BaseIIDList[] =
{
    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_BaseIIDList,
    &_tpccCom_IID_Lookup,
    2,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};
#if _MSC_VER >= 1200
#pragma warning(pop)
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AMD64) */

```

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/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;
CRITICAL_SECTION debugMutex;
CRITICAL_SECTION errorMutex;
#ifdef TIMING 1
FILE *respTimes;
struct txn
{
    short txnType;
    struct _timeb      startTime;
    struct _timeb      endTime;
    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_ordr(ordr_wrapper *ordr,void *ctx);
extern "C" TPCCDB2GLUE_API int do_dlvj(dlvj_wrapper *dlvj,void *ctx);
extern "C" TPCCDB2GLUE_API int do_stok(stok_wrapper *stok,void *ctx);

```

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/tpccDB2glue.cpp

```

// tpccDB2glue.cpp : Defines the entry point for the DLL application.
//
#include "stdafx.h"
#include "tpccDB2glue.h"
BOOL APIENTRY DllMain( HANDLE hModule,
                      DWORD ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
            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 | ios_base::app);
            if(!errorStream.rdbuf( )->is_open())
                return FALSE;
        }
    }
    #ifdef TIMING
        respTimes=fopen("c:\inetpub\wwwroot\tpcc\respTimes",
        "wb");
        if(!respTimes)
        {
            ERRORMSG("Unable to open response time
            file c:\inetpub\wwwroot\tpcc\respTimes"<<endl);
            return FALSE;
        }
        ERRORMSG("Response time file created:"<<endl);
    #endif
    DEBUGMSG("Opening registry sub key "<<
    REGISTRY_SUB_KEY << endl);
    //open up registry key
    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,REGISTER

```

```

Y_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,&
        egType,(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);

```

```

    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)
{
    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
    neword_sql(&nord->in_nord,&nord->out_nord);
    #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;
        }
        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);
    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
    payment_sql(&pymt->in_paym,&pymt->out_paym);
#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);
        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)
{
    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);
#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif
    ordstat_sql(&ords->in_ords,&ords->out_ords);
#ifdef 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);
    return OK;
}
/*
*****
** Name      :      do_dlvly
** Description :      Function calls db2 api to execute
**          :      ords txn
** Parameters :      dlvy_wrapper*  dlvy 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(dlvy_wrapper
*dlvy,void *ctx)
{
    DEBUGMSG("Entered do_dlvly, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
    int rc = attachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("dlvy failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);

```

```

        DEBUGMSG("dlvy 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);
#ifdef TIMING
    struct txn timeSample;
    _ftime(&timeSample.startTime);
#endif
    //call dlvy txn
    delivery_sql(&dlvy->in_dlvy,&dlvy->out_dlvy);
#ifdef 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,
dlvy"<<endl);
    }
    LeaveCriticalSection(&errorMutex);
#endif
    DEBUGMSG("return from delivery_sql(), s_transtatus:" <<
dlvy->out_dlvy.s_transtatus << endl);
    rc = detachContext(ctx);
    if(rc != OK)
    {
        ERRORMSG("dlvy failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        DEBUGMSG("dlvy failed detach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
        return ERR_DETACHING_CONTEXT;
    }
    DEBUGMSG("dlvy detach_context successful. dlvy txn
complete."<<endl);
    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)
{

```

```

        DEBUGMSG("Entered do_stok, attaching to context:" <<
DEBUGADDRESS(ctx) << endl);
        int rc = attachContext(ctx);
        if(rc != OK)
        {
            ERRORMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
            DEBUGMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
            return ERR_ATTACHING_CONTEXT;
        }
        DEBUGMSG("attaching to
context:"<<DEBUGADDRESS(ctx)<<" ,preparing to call db2" <<
endl);
        DEBUGMSG("calling stocklev_sql()" <<endl);
#ifdef TIMING
        struct txn timeSample;
        _ftime(&timeSample.startTime);
#endif
        //call stock level txn
        stocklev_sql(&stok->in_stok, &stok->out_stok);
#ifdef TIMING
        _ftime(&timeSample.endTime);
        timeSample.txnType=5;
        EnterCriticalSection(&errorMutex);
        if( (fwrite(&timeSample,sizeof(struct txn),1,respTimes)) !=
1 )
        {
            ERRORMSG("Unable to write to binary file,
stok"<<endl);
        }
        LeaveCriticalSection(&errorMutex);
#endif
        DEBUGMSG("return from stocklev_sql(), s_transtatus:" <<
stok->out_stok.s_transtatus << endl);
        DEBUGMSG("calling detach_context"<<endl);
        rc = detachContext(ctx);
        if(rc != OK)
        {
            ERRORMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
            DEBUGMSG("stok failed attach_context w/
ctx:"<<DEBUGADDRESS(ctx)<<" rc:" << rc << endl);
            return ERR_DETACHING_CONTEXT;
        }
        DEBUGMSG("detach_context successful. stok txn
complete."<<endl);
        return OK;
    }
}

```


Appendix - B: Tunable Parameters

B.1 Database Parameters.

db.cfg.out

Database Configuration for Database TPCC
 Database configuration release level = 0x0a00
 Database release level = 0x0a00
 Database territory = US
 Database code page = 819
 Database code set = ISO8859-1
 Database country/region code = 1
 Database collating sequence = BINARY
 Alternate collating sequence (ALT_COLLATE) =
 Dynamic SQL Query management (DYN_QUERY_MGMT) = DISABLE
 Discovery support for this database (DISCOVER_DB) = ENABLE
 Default query optimization class (DFT_QUERYOPT) = 5
 Degree of parallelism (DFT_DEGREE) = 1
 Continue upon arithmetic exceptions (DFT_SQLMATHWARN) = NO
 Default refresh age (DFT_REFRESH_AGE) = 0
 Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM
 Number of frequent values retained (NUM_FREQVALUES) = 10
 Number of quantiles retained (NUM_QUANTILES) = 20
 Backup pending = NO
 Database is consistent = 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
 Data Links Token Expiry Interval (sec) (DL_EXPINT) = 60
 Data Links Write Token Init Expiry Intvl(DL_WT_IEXPINT) = 60
 Data Links Number of Copies (DL_NUM_COPIES) = 1
 Data Links Time after Drop (days) (DL_TIME_DROP) = 1
 Data Links Token in Uppercase (DL_UPPER) = NO
 Data Links Token Algorithm (DL_TOKEN) = MACO
 Database heap (4KB) (DBHEAP) = 524288
 Size of database shared memory (4KB) (DATABASE_MEMORY) = 506600000
 Catalog cache size (4KB) (CATALOGCACHE_SZ) = (MAXAPPLS*4)
 Log buffer size (4KB) (LOGBUFSZ) = 25000
 Utilities heap size (4KB) (UTIL_HEAP_SZ) = 5000
 Buffer pool size (pages) (BUFFPAGE) = 1000
 Extended storage segments size (4KB) (ESTORE_SEG_SZ) = 16000
 Number of extended storage segments (NUM_ESTORE_SEGS) = 0
 Max storage for lock list (4KB) (LOCKLIST) = 48000

Max size of appl. group mem set (4KB) (APPGROUP_MEM_SZ) = 20000
 Percent of mem for appl. group heap (GROUPHEAP_RATIO) = 70
 Max appl. control heap size (4KB) (APP_CTL_HEAP_SZ) = 128
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = (SHEAPTHRES)
 Sort list heap (4KB) (SORTHEAP) = 16
 SQL statement heap (4KB) (STMTHEAP) = 65000
 Default application heap (4KB) (APPLHEAPSZ) = 2500
 Package cache size (4KB) (PCKCACHESZ) = 50000
 Statistics heap size (4KB) (STAT_HEAP_SZ) = 4384
 Interval for checking deadlock (ms) (DLCHKTIME) = 3000
 Percent. of lock lists per application (MAXLOCKS) = 20
 Lock timeout (sec) (LOCKTIMEOUT) = -1
 Changed pages threshold (CHNGPGS_THRESH) = 99
 Number of asynchronous page cleaners (NUM_IOCLEANERS) = 80
 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) = 800
 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 = S0000092.LOG
 Block log on disk full (BLK_LOG_DSK_FUL) = NO
 Percent of max active 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 chckpt (SOFTMAX) = 20100
 Log retain for recovery enabled (LOGRETAIN) = RECOVERY
 User exit for logging enabled (USEREXIT) = OFF
 HADR database role = STANDARD
 HADR local host name (HADR_LOCAL_HOST) =
 HADR local service name (HADR_LOCAL_SVC) =
 HADR remote host name (HADR_REMOTE_HOST) =
 HADR remote service name (HADR_REMOTE_SVC) =
 HADR instance name of remote server (HADR_REMOTE_INST) =
 HADR timeout value (HADR_TIMEOUT) = 120
 HADR log write synchronization mode (HADR_SYNCMODE) = NEARSYNC

First log archive method (LOGARCHMETH1) = LOGRETAIN
 Options for logarchmeth1 (LOGARCHOPT1) =
 Second log archive method (LOGARCHMETH2) = OFF
 Options for logarchmeth2 (LOGARCHOPT2) =
 Failover log archive path (FAILARCHPATH) =
 Number of log archive retries on error (NUMARCHRETRY) = 5
 Log archive retry Delay (secs) (ARCHRETRYDELAY) = 20
 Vendor options (VENDOROPT) =
 Auto restart enabled (AUTORESTART) = ON
 Index re-creation time and redo index build (INDEXREC) = SYSTEM (RESTART)
 Log pages during index build (LOGINDEXBUILD) = OFF
 Default number of loadrec sessions (DFT_LOADREC_SES) = 1
 Number of database backups to retain (NUM_DB_BACKUPS) = 12
 Recovery history retention (days) (REC_HIS_RETENTN) = 366
 TSM management class (TSM_MGMTCLASS) =
 TSM node name (TSM_NODENAME) =
 TSM owner (TSM_OWNER) =
 TSM password (TSM_PASSWORD) =
 Automatic maintenance (AUTO_MAINT) = OFF
 Automatic database backup (AUTO_DB_BACKUP) = OFF
 Automatic table maintenance (AUTO_TBL_MAINT) = OFF
 Automatic runstats (AUTO_RUNSTATS) = OFF
 Automatic statistics profiling (AUTO_STATS_PROF) = OFF
 Automatic profile updates (AUTO_PROF_UPD) = OFF
 Automatic reorganization (AUTO_REORG) = OFF

dbm.cfg.out

Database Manager Configuration
 Node type = Database Server with local clients
 Database manager configuration release level = 0x0a00
 CPU speed (millisec/instruction) (CPUSPEED) = 3.306409e-07
 Max number of concurrently active databases (NUMDB) = 1
 Data Links support (DATALINKS) = NO
 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) = /usr/java14_64
 Diagnostic error capture level (DIAGLEVEL) = 1
 Notify Level (NOTIFYLEVEL) = 3
 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
Database manager authentication (AUTHENTICATION) =
CLIENT
Cataloging allowed without authority (CATALOG_NOAUTH) =
YES
Trust all clients (TRUST_ALLCLNTS) = YES
Trusted client authentication (TRUST_CLNTAUTH) =
CLIENT
Bypass federated authentication (FED_NOAUTH) = NO
Default database path (DFTDBPATH) =
/home/tpcc
Database monitor heap size (4KB) (MON_HEAP_SZ) =
4096
Java Virtual Machine heap size (4KB) (JAVA_HEAP_SZ) =
1024
Audit buffer size (4KB) (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB)
(INSTANCE_MEMORY) = AUTOMATIC
Backup buffer default size (4KB) (BACKBUFSZ) = 1024
Restore buffer default size (4KB) (RESTBUFSZ) = 1024
Sort heap threshold (4KB) (SHEAPTHRES) = 20000
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
Max number of existing agents (MAXAGENTS) = 5050
Agent pool size (NUM_POOLAGENTS) = 1
Initial number of agents in pool (NUM_INITAGENTS) = 0
Max number of coordinating agents
(MAX_COORDAGENTS) = MAXAGENTS
Max no. of concurrent coordinating agents (MAXCAGENTS) =
MAX_COORDAGENTS
Max number of client connections (MAX_CONNECTIONS)
= MAX_COORDAGENTS
Keep fenced process (KEEPFENCED) = YES
Number of pooled fenced processes (FENCED_POOL) =
MAX_COORDAGENTS
Initial number of fenced processes (NUM_INITFENCED) = 0

```

```

Index re-creation time and redo index build (INDEXREC) =
RESTART
Transaction manager database name (TM_DATABASE) =
1ST_CONN
Transaction resync interval (sec) (RESYNC_INTERVAL) =
180
SPM name (SPM_NAME) =
SPM log size (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit (SPM_MAX_RESYNC) = 20
SPM log path (SPM_LOG_PATH) =
TCP/IP Service name (SVCENAME) =
db2ctpc
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)
= 512
Number of FCM request blocks (FCM_NUM_RQB) =
AUTOMATIC
Number of FCM connection entries
(FCM_NUM_CONNECT) = AUTOMATIC
Number of FCM message anchors
(FCM_NUM_ANCHORS) = AUTOMATIC

```

db2set.cfg.out

```

[i] DB2_RESOURCE_POLICY=/home/tpcc/tpc-
c.ibm/cfg/affinity.cfg
[i] DB2_SELUDI_COMM_BUFFER=Y
[i] DB2_USE_ALTERNATE_PAGE_CLEANING=YES
[i] DB2_MAX_NON_TABLE_LOCKS=1000
[i] DB2_LGPAGE_BP=YES
[i] DB2_TRUSTED_BINDIN=ON
[i] DB2_KEEPTABLELOCK=ON
[i] DB2_NO_FORK_CHECK=ON
[i] DB2_APM_PERFORMANCE=ALL
[i] DB2_ENABLE_BUFDP=OFF
[i] DB2_PINNED_BP=YES
[i] DB2_SELECTIVITY=ON
[i] DB2ASSUMEUPDATE=ON
[i] DB2CHECKCLIENTINTERVAL=0
[i] DB2_HASH_JOIN=OFF
[i] DB2CHKSQLDA=OFF
[i] DB2ENVLIST=MEMORY_AFFINITY_DATA_SEG_SPECIAL
[i] DB2_COLLECT_TS_REC_INFO=false
[i] DB2COMM=tcPIP
[i] DB2CHKPTR=OFF

```

affinity.cfg

```

<RESOURCE_POLICY>
<GLOBAL_RESOURCE_POLICY>
<METHOD>RSET</METHOD>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00000</RESOURCE>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00001</RESOURCE>

```

```

</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00002</RESOURCE>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00003</RESOURCE>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00004</RESOURCE>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00005</RESOURCE>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00006</RESOURCE>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00007</RESOURCE>
</RESOURCE_BINDING>
</GLOBAL_RESOURCE_POLICY>
</RESOURCE_POLICY>

```

B.2 Transaction Monitor Parameters

tpccCom.tpcc.com.settings.txt

```

Transactions not supported
Enable object pooling
Minimum pool size 24
Maximum pool size 24
Creation timeout 100,000
Enable Object Construction
Enable Just in time activation
Concurrency Required

```

InetInfo registry.reg

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\InetInfo\Parameters]
"ListenBackLog"=dword:000000fa
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,
43,00,00,00,00,00
"MaxConnections"=dword:000061a8
"PoolThreadLimit"=dword:00000190
"ThreadTimeout"=dword:00015180
"MaxConcurrency"=dword:ffffff

```

tcPIP parameters registry.reg

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\TcPIP\Parameters]

```

"NV Hostname"="client32"
 "DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\

00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\

64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
 "NameServer"=""
 "ForwardBroadcasts"=dword:00000000
 "IPEnableRouter"=dword:00000000
 "Domain"=""
 "Hostname"="client32"
 "SearchList"=""
 "UseDomainNameDevolution"=dword:00000001
 "EnableICMPRedirect"=dword:00000001
 "DeadGWDetectDefault"=dword:00000001
 "DontAddDefaultGatewayDefault"=dword:00000000
 "EnableSecurityFilters"=dword:00000000
 "AllowUnqualifiedQuery"=dword:00000000
 "PrioritizeRecordData"=dword:00000001
 "GlobalTcpWindowSize"=dword:00008000
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters]
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\NdisWanlp]
 "LLInterface"="WANARP"
 "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,00,31,00,44,00,33,00,30,39,00,\

37,00,37,00,2d,00,37,00,31,00,30,00,46,00,2d,00,34,00,41,00,37,00,45,00,2d,\

00,38,00,44,00,34,00,37,00,2d,00,34,00,42,00,34,00,38,00,32,00,42,00,43,00,\

35,00,32,00,33,00,46,00,38,00,7d,00,00,00,54,00,63,00,70,00,69,00,70,00,5c,00,\

00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,\

6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,00,39,00,41,\

00,33,00,41,00,41,00,36,00,41,00,43,00,2d,00,35,00,38,00,34,00,36,00,2d,00,\

34,00,30,00,37,00,45,00,2d,00,38,00,32,00,35,00,30,00,2d,00,46,00,30,00,33,\

00,42,00,36,00,30,00,34,00,39,00,36,00,36,00,44,00,43,00,7d,00,00,00,00,00
 "NumInterfaces"=dword:00000002

"IpInterfaces"=hex:77,39,d3,61,0f,71,7e,4a,8d,47,4b,48,2b,c5,23,f8,ac,a6,3a,9a,\

46,58,7e,40,82,50,f0,3b,60,49,66,dc
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{0435C97F-9186-473F-B181-5449A2CF0042}]
 "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,00,30,00,34,00,33,00,35,00,43,00,39,00,\

37,00,46,00,2d,00,39,00,31,00,38,00,36,00,2d,00,34,00,37,00,33,00,46,00,2d,\

00,42,00,31,00,38,00,31,00,2d,00,35,00,34,00,34,00,39,00,41,00,32,00,43,00,\

46,00,30,00,30,00,34,00,32,00,7d,00,00,00,00,00
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{1E07A95A-92A0-4836-BF73-7AE38F8ACA07}]
 "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,00,31,00,45,00,30,00,37,00,04,00,39,00,\

35,00,41,00,2d,00,39,00,32,00,41,00,30,00,2d,00,34,00,38,00,33,00,36,00,2d,\

00,42,00,46,00,37,00,33,00,2d,00,37,00,41,00,45,00,33,00,38,00,46,00,38,00,\

41,00,43,00,41,00,30,00,37,00,7d,00,00,00,00,00
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{2EA04AA5-93A6-437F-9153-2F6834D3B795}]
 "LLInterface"=""
 "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00,45,00,41,00,30,00,34,00,41,00,\

41,00,35,00,2d,00,39,00,33,00,41,00,36,00,2d,00,34,00,33,00,37,00,46,00,2d,\

00,39,00,31,00,35,00,33,00,2d,00,32,00,46,00,36,00,38,00,33,00,34,00,44,00,\

33,00,42,00,37,00,39,00,35,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{37430121-7BE3-4B55-8AAB-D8AD09B2029C}]
 "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,00,33,00,37,00,34,00,33,00,30,30,31,00,\

32,00,31,00,2d,00,37,00,42,00,45,00,33,00,2d,00,34,00,42,00,35,00,35,00,2d,\

00,38,00,41,00,41,00,42,00,2d,00,44,00,38,00,41,00,44,00,30,00,39,00,42,00,\

32,00,30,00,32,00,39,00,43,00,7d,00,00,00,00,00
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{6FE29D81-59D5-4401-A77E-BE3BC929B6E0}]
 "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,00,36,00,46,00,45,00,32,00,39,00,44,00,\

38,00,31,00,2d,00,35,00,39,00,44,00,35,00,2d,00,34,00,34,00,30,00,31,00,2d,\

00,41,00,37,00,37,00,45,00,2d,00,42,00,45,00,33,00,42,00,43,00,39,00,32,00,\

39,00,42,00,36,00,45,00,30,00,7d,00,00,00,00,00
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{7B215199-A3F3-4836-89A6-390C5E70E801}]
 "LLInterface"=""
 "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,37,00,42,00,32,00,31,00,35,00,31,00,\

39,00,39,00,2d,00,41,00,33,00,46,00,33,00,2d,00,34,00,38,00,33,00,36,00,2d,\

00,38,00,39,00,41,00,36,00,2d,00,33,00,39,00,30,00,43,00,35,00,45,00,37,00,\

30,00,45,00,38,00,30,00,31,00,7d,00,00,00,00,00
 [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{A32BB4A3-C9B2-4ADB-A65D-18BB314BF7F0}]
 "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,00,41,00,33,00,32,00,42,00,42,00,34,00,\

41,00,33,00,2d,00,43,00,39,00,42,00,32,00,2d,00,34,00,41,00,44,00,42,00,2d,\

00,41,00,36,00,35,00,44,00,2d,00,31,00,38,00,42,00,42,00,33,00,31,00,34,00,\

42,00,46,00,37,00,46,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{A71EB7B5-37C6-42DB-BE8F-BB231FD1BE00}]

"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,00,41,00,37,00,31,00,45,00,42,00,37,00,\

42,00,35,00,2d,00,33,00,37,00,43,00,36,00,2d,00,34,00,32,00,44,00,42,00,2d,\

00,42,00,45,00,38,00,46,00,2d,00,42,00,42,00,32,00,33,00,31,00,46,00,44,00,\

31,00,42,00,45,00,30,00,30,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{BEABCC14-9C0A-4BE9-9817-14C4092418D3}]

"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,00,42,00,45,00,41,00,42,00,43,00,43,00,\

31,00,34,00,2d,00,39,00,43,00,30,00,41,00,2d,00,34,00,42,00,45,00,39,00,2d,\

00,39,00,38,00,31,00,37,00,2d,00,31,00,34,00,43,00,34,00,30,00,39,00,32,00,\

34,00,31,00,38,00,44,00,33,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Adapters\{CD3F7746-9E60-4E22-9A40-7BC6CC6B2E2E}]

"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,00,43,00,44,00,33,00,46,00,37,00,37,00,\

34,00,36,00,2d,00,39,00,45,00,36,00,30,00,2d,00,34,00,45,00,32,00,32,00,2d,\

00,39,00,41,00,34,00,30,00,2d,00,37,00,42,00,43,00,36,00,43,00,43,00,36,00,\

42,00,32,00,45,00,32,00,45,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\DNSRegisteredAdapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{0435C97F-9186-473F-B181-5449A2CF0042}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000000

"IPAddress"=hex(7):31,00,33,00,35,00,2e,00,31,00,2e,00,31,00,2e,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

"DefaultGateway"=hex(7):00,00

"DefaultGatewayMetric"=hex(7):00,00

"NameServer"=""

"Domain"=""

"DisableDynamicUpdate"=dword:00000000

"EnableAdapterDomainNameRegistration"=dword:00000000

"InterfaceMetric"=dword:00000001

"TCPAllowedPorts"=hex(7):30,00,00,00,00,00

"UDPAllowedPorts"=hex(7):30,00,00,00,00,00

"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00

"NTEContextList"=hex(7):00,00

"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{1E07A95A-92A0-4836-BF73-7AE38F8ACA07}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000001

"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,\

00,00,00,00

"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,2e,00,30,\

00,00,00,00,00

"DefaultGateway"=hex(7):00,00

"DefaultGatewayMetric"=hex(7):00,00

"NameServer"=""

"Domain"=""

"DisableDynamicUpdate"=dword:00000000

"EnableAdapterDomainNameRegistration"=dword:00000000

"InterfaceMetric"=dword:00000001

"TCPAllowedPorts"=hex(7):30,00,00,00,00,00

"UDPAllowedPorts"=hex(7):30,00,00,00,00,00

"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00

"NTEContextList"=hex(7):00,00

"DhcpIPAddress"="0.0.0.0"

"DhcpSubnetMask"="255.0.0.0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{2EA04AA5-93A6-437F-9153-2F6834D3B795}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000000

"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,36,00,38,00,2e,00,31,00,31,00,\

2e,00,35,00,31,00,00,00,00,00

"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\

00,2e,00,32,00,35,\

00,35,00,2e,00,30,00,00,00,00,00

"DefaultGateway"=hex(7):00,00

"DefaultGatewayMetric"=hex(7):00,00

"NameServer"=""

"Domain"=""

"DisableDynamicUpdate"=dword:00000000

"EnableAdapterDomainNameRegistration"=dword:00000000

"InterfaceMetric"=dword:00000001

"TCPAllowedPorts"=hex(7):30,00,00,00,00,00

"UDPAllowedPorts"=hex(7):30,00,00,00,00,00

"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00

"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\

33,00,00,00,00,00

"DhcpServer"="255.255.255.255"

"Lease"=dword:00000e10

"LeaseObtainedTime"=dword:40b39c06

"T1"=dword:40b3a30e

"T2"=dword:40b3a854

"LeaseTerminatesTime"=dword:40b3aa16

"IPAutoconfigurationAddress"="0.0.0.0"

"IPAutoconfigurationMask"="255.255.0.0"

"IPAutoconfigurationSeed"=dword:00000000

"AddressType"=dword:00000000

"DhcpClassIdBin"=hex:

"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{37430121-7BE3-4B55-8AAB-D8AD09B2029C}]

"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001

"EnableDHCP"=dword:00000001

"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,\

00,00,00,00

"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,2e,00,30,\

00,00,00,00,00

"DefaultGateway"=hex(7):00,00

"DefaultGatewayMetric"=hex(7):00,00

"NameServer"=""

"Domain"=""

"DisableDynamicUpdate"=dword:00000000

"EnableAdapterDomainNameRegistration"=dword:00000000

"InterfaceMetric"=dword:00000001

"TCPAllowedPorts"=hex(7):30,00,00,00,00,00

"UDPAllowedPorts"=hex(7):30,00,00,00,00,00

"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00

"NTEContextList"=hex(7):00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{61D33977-710F-4A7E-8D47-4B482BC523F8}]

"UseZeroBroadcast"=dword:00000000

"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{6FE29D81-59D5-4401-A77E-BE3BC929B6E0}]
"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{7B215199-A3F3-4836-89A6-390C5E70E801}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{9A3AA6AC-5846-407E-8250-F03B604966DC}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001

"DontAddDefaultGateway"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{A32BB4A3-C9B2-4ADB-A65D-18BB314BF7F0}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,33,00,2e,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,00,32,00,35,00,35,00,35,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{A71EB7B5-37C6-42DB-BE8F-BB231FD1BE00}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,33,00,32,00,2e,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,00,32,00,35,00,35,00,35,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:40be0640
"T1"=dword:40be0d48
"T2"=dword:40be128e
"LeaseTerminatesTime"=dword:40be1450
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{BEABCC14-90CA-4BE9-9817-14C4092418D3}]

"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,32,00,2e,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
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Interfaces\{CD3F7746-9E60-4E22-9A40-7BC6CC6B2E2E}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
"DhcpIPAddress"="0.0.0.0"
"DhcpSubnetMask"="255.0.0.0"
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\PersistentRoutes]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service s\Tcpip\Parameters\Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,00,6c,00,6c,00,00,00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,00,06,00,00,00,02,\

```

00,00,00,01,00,00,00,00,00,00,02,00,00,00,00,00,00,06,0
0,00,00,00,00,\
00,00,00,00,00,06,00,00,00,00,00,00,01,00,00,00,06,00,0
0,00,02,00,00,\
00,02,00,00,00,11,00,00,00,02,00,00,02,00,00,00,00,00,0
0,02,00,00,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,11,00,00,00,0
0,00,00,00,02,\
00,00,00,11,00,00,00,02,00,00,03,00,00,00,00,00,00,00

```

Tpcc software registry.reg

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"dlvyLogPath"="c:\inetpub\wwwroot\tpcc\dlvy"
"dlvyQueueLen"=dword:00004e20
"nullDB"=dword:00000000
"htmlTrace"=dword:00000000
"dbName"="tpcc"
"errorLogFile"="c:\inetpub\wwwroot\tpcc\isapi_err.log"
"htmlTraceLogFile"="c:\inetpub\wwwroot\tpcc\isapi_log"
"numUsers"=dword:00005208
"dbType"="DB2"
"dbUserName"="tpcc"
"dbPassword"="tpcc"
"dbInterfacePath"="C:\inetpub\wwwroot\tpcc\db2glue.dll"
"dlvyThreads"=dword:00000005
"isapi_trace"=dword:00000000

```

W3SVC registry.reg

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00,
54,00,5c,00,53,00,\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,6
5,00,74,00,73,\
00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,0
0,6f,00,2e,00,\
65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,00,4d,0
0,49,00,4e,00,00,00,\
00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"

```

```

"Description"="Provides Web connectivity and administration
through the Internet Information Services snap-in."
"FailureActions"=hex:ff,ff,ff,ff,00,00,00,00,00,00,00,03,00,00,
00,38,c3,0f,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\ASP\Parameters]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\WINNT\System32\inetrv"
"CertMapList"="C:\WINNT\System32\inetrv\iisrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\WINNT\System32\LogFiles"
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\ADCLaunch]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\Script Map]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Parameters\Virtual Roots]
"/"="c:\inetpub\wwwroot,,201"
"/Scripts"="c:\inetpub\scripts,,204"
"/IISHelp"="c:\winnt\help\iishelp,,201"
"/IISAdmin"="C:\WINNT\System32\inetrv\iisadmin,,201"
"/IISamples"="c:\inetpub\iisamples,,201"
"/MSADC"="c:\program files\common
files\system\msadc,,205"
"/_vti_bin"="C:\Program Files\Common Files\Microsoft
Shared\Web Server Extensions\40\isapi,,205"
"/Printers"="C:\WINNT\web\printers,,201"
"/tpcc"="C:\inetpub\wwwroot\tpcc,,207"
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Performance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008e6
"Last Help"=dword:000008e7
"First Counter"=dword:00000844
"First Help"=dword:00000845
"Library Validation
Code"=hex:4a,82,e5,79,9f,41,c4,01,10,3d,00,00,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,9a,a9,c0,5f,3f,c4,01
"WbemAdapFileSize"=dword:00003d10
"WbemAdapStatus"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,0
0,30,00,00,00,02,\

```

```

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,
00,01,00,00,\
00,00,02,00,70,00,04,00,00,00,00,18,00,fd,01,02,00,01,01,0
0,00,00,00,\
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,
00,00,00,05,\
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,00,
1,01,00,00,00,\
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,0
2,00,00,00,00,\
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,0
0,05,12,00,00,\
00,01,01,00,00,00,00,00,05,12,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service
s\W3SVC\Enum]
"0"="Root\LEGACY_W3SVC\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

```

B.3 AIX Parameters

IBM eServer p5 595

Isattr -El sys0		
SW_dist_intr	false	Enable SW distribution of interrupts
True		
autorestart	true	Automatically REBOOT system 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		
ent_capacity	64.00	Entitled processor capacity
False		
frequency	0	System Bus Frequency
False		
fullcore	false	Enable full CORE dump
True		
fwversion	IBM,SF222_043	Firmware version and revision
levels	False	
id_to_partition	0X80000849D9800002	Partition ID
False		

```

id_to_system 0X80000849D9800000 System ID
False
iostat      false      Continuously maintain DISK I/O history
True
keylock     normal      State of system keylock at boot
time       False
max_capacity 64.00      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 MBUFFS True
maxpout     511         HIGH water mark for pending write
I/Os per file True
maxuproc    20000      Maximum number of
PROCESSES allowed per user True
min_capacity 1.00      Minimum potential processor
capacity   False
minpout     384         LOW water mark for pending write
I/Os per file True
modelname   IBM,9119-595    Machine name
ncargs      6          ARG/ENV list size in 4K byte blocks
True
pre430core  false      Use pre-430 style CORE dump
True
pre520tune  disable     Pre-520 tuning compatibility mode
True
realmem     2135949312    Amount of usable physical
memory in Kbytes False
rtasversion 1          Open Firmware RTAS version
False
systemid    IBM,020256A3C   Hardware system identifier
False
variable_weight 0      Variable processor capacity weight
False
vmo -L
NAME        CUR  DEF  BOOT  MIN  MAX  UNIT
TYPE
DEPENDENCIES
-----
memory_frames 509M 509M 4KB pages
S
-----
pinnable_frames 10070K 10070K 4KB
pages S
-----
maxfree 128 128 128 16 200K 4KB pages
D
  minfree
  memory_frames
-----
minfree 120 120 120 8 200K 4KB pages
D
  maxfree
  memory_frames
-----
minperm% 20 20 20 1 100 % memory
D
  maxperm%

```

```

-----
minperm 2128K 2128K S
-----
maxperm% 80 80 80 1 100 % memory
D
  minperm%
  maxclient%
-----
maxperm 8514K 8514K S
-----
strict_maxperm 0 0 0 0 1 boolean
D
-----
maxpin% 98 80 98 1 99 % memory
D
  pinnable_frames
  memory_frames
-----
maxpin 520962K 520962K
S
-----
maxclient% 80 80 80 1 100 % memory
D
  maxperm%
-----
lrubucket 128K 128K 128K 64K 4KB pages
D
-----
defps 1 1 1 0 1 boolean D
-----
nokilluid 0 0 0 0 4G-1 uid D
-----
numpsblks 26048K 26048K 4KB pages
S
-----
npskill 208384 208384 208384 1 25M-1 4KB
pages D
-----
npswam 814K 814K 814K 0 25M-1 4KB
pages D
-----
v_pinshm 1 0 1 0 1 boolean D
-----
pta_balance_threshold n/a 1 1 0 99 % pta
segment D
-----
pagecoloring n/a 0 0 0 1 boolean B
-----
framesets 2 2 2 1 10 B
-----
mempools 1 1 1 1 128 B
-----
lgpg_size 16M 0 16M 0 16M bytes
D
  lgpg_regions
-----
lgpg_regions 124000 0 124000 0
D
  lgpg_size
-----
num_spec_dataseg 0 0 0 0 B
-----

```

```

spec_dataseg_int 512 512 512 0
B
-----
memory_affinity 1 1 1 0 1 boolean
B
-----
htabscale n/a -1 -1 -4 0 B
-----
force_realias_lite 0 0 0 0 1 boolean D
-----
realias_percentage 0 0 0 0 32K-1
D
-----
rpgcontrol 2 2 2 0 3 D
-----
rpgclean 0 0 0 0 1 boolean D
-----
npsrpgmin 1221K 1221K 1221K 0 25M-1 4KB
pages D
  npsrpgmax
-----
npsrpgmax 1628K 1628K 1628K 0 25M-1 4KB
pages D
  npsrpgmin
-----
scrub 0 0 0 0 1 boolean D
-----
scrubclean 0 0 0 0 1 boolean D
-----
npsscubmin 1221K 1221K 1221K 0 25M-1 4KB
pages D
  npsscubmax
-----
npsscubmax 1628K 1628K 1628K 0 25M-1 4KB
pages D
  npsscubmin
-----
data_stagger_interval 161 161 161 0 4K-1 4KB
pages D
  lgpg_regions
-----
large_page_heap_size 0 0 0 0 8E-1 bytes
B
  lgpg_regions
-----
kernel_heap_psize 4K 4K 4K 4K 16M bytes
B
  lgpg_regions
-----
soft_min_lgpgs_vmpool 0 0 0 0 90 %
D
  lgpg_regions
-----
vm_modlist_threshold -1 -1 -1 -2 2G-1
D
-----
vmm_fork_policy 1 1 1 0 1 boolean
D
-----
low_ps_handling 1 1 1 1 2 D
-----

```

mbuf_heap_psize 4K 4K 4K 4K 16M bytes
B

strict_maxclient 1 1 1 0 1 D

cpu_scale_memp 8 8 8 1 64 B

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

Value conventions:

K = Kilo: 2¹⁰ G = Giga: 2³⁰ P = Peta: 2⁵⁰

M = Mega: 2²⁰ T = Tera: 2⁴⁰ E = Exa: 2⁶⁰

ioo -L

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT
TYPE						

DEPENDENCIES

minpgahead 2 2 2 0 4K 4KB pages
D

maxpgahead

maxpgahead 8 8 8 0 4K 4KB pages
D

minpgahead

pd_npages 64K 64K 64K 1 512K 4KB
pages D

maxrandwrt 0 0 0 0 512K 4KB pages
D

numclust 1 1 1 0 2G-1 16KB/cluster
D

numfsbufs 196 196 196 1 2G-1
M

sync_release_ilock 0 0 0 0 1 boolean
D

lvm_bufcnt 9 9 9 1 64 128KB/buffer
D

j2_minPageReadAhead 2 2 2 0 64K 4KB
pages D

j2_maxPageReadAhead 128 128 128 0 64K 4KB
pages D

j2_nBufferPerPagerDevice 512 512 512 0 256K
M

j2_nPagesPerWriteBehindCluster
32 32 32 0 64K D

j2_maxRandomWrite 0 0 0 0 64K 4KB
pages D

j2_nRandomCluster 0 0 0 0 64K 16KB
clusters D

jfs_cread_enabled 0 0 0 0 1 boolean
D

jfs_use_read_lock 1 1 1 0 1 boolean
D

j2_inodeCacheSize 400 400 400 1 1000
D

j2_metadataCacheSize 400 400 400 1 1000
D

pv_min_pbuf 512 512 512 512 2G-1
D

j2_dynamicBufferPreallocation
16 16 16 0 256 16k slabs M

j2_maxUsableMaxTransfer 512 512 512 1 4K
pages M

j2_non_fatal_crashes_system
0 0 0 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

Value conventions:

K = Kilo: 2¹⁰ G = Giga: 2³⁰ P = Peta: 2⁵⁰

M = Mega: 2²⁰ T = Tera: 2⁴⁰ E = Exa: 2⁶⁰

Appendix - C: Database Setup Code

C.1 Database Creation Scripts

db/create_database.ddl

```
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/rD1F01V1CSTI' 319872
)
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/rD1F01V2CSTI' 319872
)
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/rD1F01V3CSTI' 319872
)
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/rD1F01V4CSTI' 319872
)
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/rD1F02V1CSTI' 319872
)
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/rD1F02V2CSTI' 319872
)
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/rD1F02V3CSTI' 319872
)
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/rD1F02V4CSTI' 319872
)
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/rD1F03V1CSTI' 319872
)
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/rD1F03V2CSTI' 319872
)
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/rD1F03V3CSTI' 319872
)
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/rD1F03V4CSTI' 319872
)
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/rD1F04V1CSTI' 319872
)
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/rD1F04V2CSTI' 319872
)
```

```

    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/rD1F04V3CSTI' 319872
)
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/rD1F04V4CSTI' 319872
)
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/rD1F05V1CSTI' 319872
)
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/rD1F05V2CSTI' 319872
)
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/rD1F05V3CSTI' 319872
)
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/rD1F05V4CSTI' 319872
)
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/rD1F06V1CSTI' 319872
)
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/rD1F06V2CSTI' 319872
)
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/rD1F06V3CSTI' 319872
)
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/rD1F06V4CSTI' 319872
)
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/rD1F07V1CSTI' 319872
)
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/rD1F07V2CSTI' 319872
)
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/rD1F07V3CSTI' 319872
)
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/rD1F07V4CSTI' 319872
)
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/rD1F08V1CSTI' 319872
)
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/rD1F08V2CSTI' 319872
)
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/rD1F08V3CSTI' 319872
)
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/rD1F08V4CSTI' 319872
)
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/rD1F09V1CSTI' 319872
)
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/rD1F09V2CSTI' 319872
)
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/rD1F09V3CSTI' 319872
)
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/rD1F09V4CSTI' 319872
)
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/rD1F10V1CSTI' 319872
)
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/rD1F10V2CSTI' 319872
)
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/rD1F10V3CSTI' 319872
)
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/rD1F10V4CSTI' 319872
)
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/rD1F11V1CSTI' 319872
)
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/rD1F11V2CSTI' 319872
)
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/rD1F11V3CSTI' 319872
)
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/rD1F11V4CSTI' 319872
)
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/rD1F12V1CSTI' 319872
)

```

```

    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/rD1F12V2CSTI' 319872
)
    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/rD1F12V3CSTI' 319872
)
    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/rD1F12V4CSTI' 319872
)
    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/rD1F13V1CSTI' 319872
)
    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/rD1F13V2CSTI' 319872
)
    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/rD1F13V3CSTI' 319872
)
    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/rD1F13V4CSTI' 319872
)
    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/rD1F14V1CSTI' 319872
)
    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/rD1F14V2CSTI' 319872
)
    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/rD1F14V3CSTI' 319872
)
    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/rD1F14V4CSTI' 319872
)
    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/rD1F15V1CSTI' 319872
)
    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/rD1F15V2CSTI' 319872
)
    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/rD1F15V3CSTI' 319872
)
    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/rD1F15V4CSTI' 319872
)
    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/rD1F16V1CSTI' 319872
)
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/rD1F16V2CSTI' 319872
)
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/rD1F16V3CSTI' 319872
)
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/rD1F16V4CSTI' 319872
)
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/rD1F17V1CSTI' 319872
)
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/rD1F17V2CSTI' 319872
)
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/rD1F17V3CSTI' 319872
)
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/rD1F17V4CSTI' 319872
)
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/rD1F18V1CSTI' 319872
)
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/rD1F18V2CSTI' 319872
)
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/rD1F18V3CSTI' 319872
)
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/rD1F18V4CSTI' 319872
)
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/rD1F19V1CSTI' 319872
)
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/rD1F19V2CSTI' 319872
)
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/rD1F19V3CSTI' 319872
)
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/rD1F19V4CSTI' 319872
)

```

```

    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/rD1F20V1CSTI' 319872
)
    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/rD1F20V2CSTI' 319872
)
    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/rD1F20V3CSTI' 319872
)
    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/rD1F20V4CSTI' 319872
)
    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/rD1F21V1CSTI' 319872
)
    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/rD1F21V2CSTI' 319872
)
    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/rD1F21V3CSTI' 319872
)
    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/rD1F21V4CSTI' 319872
)
    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/rD1F22V1CSTI' 319872
)
    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/rD1F22V2CSTI' 319872
)
    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/rD1F22V3CSTI' 319872
)
    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/rD1F22V4CSTI' 319872
)
    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/rD1F23V1CSTI' 319872
)
    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/rD1F23V2CSTI' 319872
)
    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/rD1F23V3CSTI' 319872
)
    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/rD1F23V4CSTI' 319872
  )
  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/rD1F24V1CSTI' 319872
  )
  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/rD1F24V2CSTI' 319872
  )
  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/rD1F24V3CSTI' 319872
  )
  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/rD1F24V4CSTI' 319872
  )
  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/rD1F25V1CSTI' 319872
  )
  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/rD1F25V2CSTI' 319872
  )
  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/rD1F25V3CSTI' 319872
  )
  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/rD1F25V4CSTI' 319872
  )
  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/rD1F26V1CSTI' 319872
  )
  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/rD1F26V2CSTI' 319872
  )
  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/rD1F26V3CSTI' 319872
  )
  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/rD1F26V4CSTI' 319872
  )
  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/rD1F27V1CSTI' 319872
  )
  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/rD1F27V2CSTI' 319872
  )
  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/rD1F27V3CSTI' 319872
  )

```

```

    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/rD1F27V4CSTI' 319872
)
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/rD1F28V1CSTI' 319872
)
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/rD1F28V2CSTI' 319872
)
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/rD1F28V3CSTI' 319872
)
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/rD1F28V4CSTI' 319872
)
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/rD1F29V1CSTI' 319872
)
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/rD1F29V2CSTI' 319872
)
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/rD1F29V3CSTI' 319872
)
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/rD1F29V4CSTI' 319872
)
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/rD1F30V1CSTI' 319872
)
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/rD1F30V2CSTI' 319872
)
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/rD1F30V3CSTI' 319872
)
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/rD1F30V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_121 of D1
drop tablespace is_customer_121;
create regular tablespace is_customer_121 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_122 of D1
drop tablespace is_customer_122;
create regular tablespace is_customer_122 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_123 of D1
drop tablespace is_customer_123;

```



```

create regular tablespace is_customer_123 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_124 of D1
drop tablespace is_customer_124;
create regular tablespace is_customer_124 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_125 of D1
drop tablespace is_customer_125;
create regular tablespace is_customer_125 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_126 of D1
drop tablespace is_customer_126;
create regular tablespace is_customer_126 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_127 of D1
drop tablespace is_customer_127;
create regular tablespace is_customer_127 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_128 of D1
drop tablespace is_customer_128;
create regular tablespace is_customer_128 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F32V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_129 of D1
drop tablespace is_customer_129;
create regular tablespace is_customer_129 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_130 of D1
drop tablespace is_customer_130;
create regular tablespace is_customer_130 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_131 of D1
drop tablespace is_customer_131;
create regular tablespace is_customer_131 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_132 of D1
drop tablespace is_customer_132;
create regular tablespace is_customer_132 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_133 of D1
drop tablespace is_customer_133;
create regular tablespace is_customer_133 pagesize 8K
managed by database
using
(

```

```

    device '/dev/rD1F34V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_134 of D1
drop tablespace is_customer_134;
create regular tablespace is_customer_134 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_135 of D1
drop tablespace is_customer_135;
create regular tablespace is_customer_135 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_136 of D1
drop tablespace is_customer_136;
create regular tablespace is_customer_136 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_137 of D1
drop tablespace is_customer_137;
create regular tablespace is_customer_137 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_138 of D1
drop tablespace is_customer_138;
create regular tablespace is_customer_138 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V2CSTI' 319872
)

```

```

    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_139 of D1
drop tablespace is_customer_139;
create regular tablespace is_customer_139 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_140 of D1
drop tablespace is_customer_140;
create regular tablespace is_customer_140 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_141 of D1
drop tablespace is_customer_141;
create regular tablespace is_customer_141 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_142 of D1
drop tablespace is_customer_142;
create regular tablespace is_customer_142 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_143 of D1
drop tablespace is_customer_143;
create regular tablespace is_customer_143 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V3CSTI' 319872
)
extentsize 256
prefetchsize 4096

```

```

    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_144 of D1
drop tablespace is_customer_144;
create regular tablespace is_customer_144 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_145 of D1
drop tablespace is_customer_145;
create regular tablespace is_customer_145 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_146 of D1
drop tablespace is_customer_146;
create regular tablespace is_customer_146 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_147 of D1
drop tablespace is_customer_147;
create regular tablespace is_customer_147 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_148 of D1
drop tablespace is_customer_148;
create regular tablespace is_customer_148 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_149 of D1
drop tablespace is_customer_149;
create regular tablespace is_customer_149 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_150 of D1
drop tablespace is_customer_150;
create regular tablespace is_customer_150 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V2CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_151 of D1
drop tablespace is_customer_151;
create regular tablespace is_customer_151 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V3CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_152 of D1
drop tablespace is_customer_152;
create regular tablespace is_customer_152 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V4CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_153 of D1
drop tablespace is_customer_153;
create regular tablespace is_customer_153 pagesize 8K
managed by database
using
(
    device '/dev/rD1F39V1CSTI' 319872
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_154 of D1
drop tablespace is_customer_154;

```

```

create regular tablespace is_customer_154 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F39V2CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_155 of D1
drop tablespace is_customer_155;
create regular tablespace is_customer_155 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F39V3CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_156 of D1
drop tablespace is_customer_156;
create regular tablespace is_customer_156 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F39V4CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_157 of D1
drop tablespace is_customer_157;
create regular tablespace is_customer_157 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F40V1CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_158 of D1
drop tablespace is_customer_158;
create regular tablespace is_customer_158 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F40V2CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_159 of D1
drop tablespace is_customer_159;
create regular tablespace is_customer_159 pagesize 8K
  managed by database

```

```

  using
  (
    device '/dev/rD1F40V3CSTI' 319872
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_customer_160 of D1
drop tablespace is_customer_160;
create regular tablespace is_customer_160 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F40V4CSTI' 319872
  )
  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' 258368
  )
  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' 258368
  )
  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' 258368
  )
  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' 258368
  )
  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/rD1F02V1ORDI' 258368
  )
  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/rD1F02V2ORDI' 258368
  )
  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/rD1F02V3ORDI' 258368
  )
  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/rD1F02V4ORDI' 258368
  )
  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/rD1F03V1ORDI' 258368
)
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/rD1F03V2ORDI' 258368
)
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/rD1F03V3ORDI' 258368
)
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/rD1F03V4ORDI' 258368
)
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/rD1F04V1ORDI' 258368
)
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/rD1F04V2ORDI' 258368
)
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/rD1F04V3ORDI' 258368
)
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/rD1F04V4ORDI' 258368
)
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/rD1F05V1ORDI' 258368
)
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/rD1F05V2ORDI' 258368
)
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/rD1F05V3ORDI' 258368
)
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/rD1F05V4ORDI' 258368
)
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/rD1F06V1ORDI' 258368
)
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/rD1F06V2ORDI' 258368
)
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/rD1F06V3ORDI' 258368
)
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/rD1F06V4ORDI' 258368
)

```

```

    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/rD1F07V1ORDI' 258368
)
    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/rD1F07V2ORDI' 258368
)
    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/rD1F07V3ORDI' 258368
)
    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/rD1F07V4ORDI' 258368
)
    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/rD1F08V1ORDI' 258368
)
    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/rD1F08V2ORDI' 258368
)
    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/rD1F08V3ORDI' 258368
)
    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/rD1F08V4ORDI' 258368
)
    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/rD1F09V1ORDI' 258368
)
    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/rD1F09V2ORDI' 258368
)
    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/rD1F09V3ORDI' 258368
)
    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/rD1F09V4ORDI' 258368
)
    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/rD1F10V1ORDI' 258368
)
    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/rD1F10V2ORDI' 258368
)
    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/rD1F10V3ORDI' 258368
)
    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/rD1F10V4ORDI' 258368
  )
  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/rD1F11V1ORDI' 258368
  )
  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/rD1F11V2ORDI' 258368
  )
  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/rD1F11V3ORDI' 258368
  )
  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/rD1F11V4ORDI' 258368
  )
  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/rD1F12V1ORDI' 258368
  )
  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/rD1F12V2ORDI' 258368
  )
  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/rD1F12V3ORDI' 258368
  )
  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/rD1F12V4ORDI' 258368
  )
  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/rD1F13V1ORDI' 258368
  )
  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/rD1F13V2ORDI' 258368
  )
  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/rD1F13V3ORDI' 258368
  )
  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/rD1F13V4ORDI' 258368
  )
  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/rD1F14V1ORDI' 258368
  )
  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/rD1F14V2ORDI' 258368
  )
  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/rD1F14V3ORDI' 258368
  )

```

```

    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/rD1F14V4ORDI' 258368
)
    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/rD1F15V1ORDI' 258368
)
    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/rD1F15V2ORDI' 258368
)
    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/rD1F15V3ORDI' 258368
)
    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/rD1F15V4ORDI' 258368
)
    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/rD1F16V1ORDI' 258368
)
    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/rD1F16V2ORDI' 258368
)
    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/rD1F16V3ORDI' 258368
)
    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/rD1F16V4ORDI' 258368
)
    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/rD1F17V1ORDI' 258368
)
    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/rD1F17V2ORDI' 258368
)
    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/rD1F17V3ORDI' 258368
)
    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/rD1F17V4ORDI' 258368
)
    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/rD1F18V1ORDI' 258368
)
    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/rD1F18V2ORDI' 258368
)
    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/rD1F18V3ORDI' 258368
)
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/rD1F18V4ORDI' 258368
)
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/rD1F19V1ORDI' 258368
)
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/rD1F19V2ORDI' 258368
)
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/rD1F19V3ORDI' 258368
)
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/rD1F19V4ORDI' 258368
)
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/rD1F20V1ORDI' 258368
)
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/rD1F20V2ORDI' 258368
)
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/rD1F20V3ORDI' 258368
)
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/rD1F20V4ORDI' 258368
)
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/rD1F21V1ORDI' 258368
)
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/rD1F21V2ORDI' 258368
)
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/rD1F21V3ORDI' 258368
)
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/rD1F21V4ORDI' 258368
)
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/rD1F22V1ORDI' 258368
)
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/rD1F22V2ORDI' 258368
)

```



```

    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/rD1F22V3ORDI' 258368
)
    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/rD1F22V4ORDI' 258368
)
    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/rD1F23V1ORDI' 258368
)
    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/rD1F23V2ORDI' 258368
)
    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/rD1F23V3ORDI' 258368
)
    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/rD1F23V4ORDI' 258368
)
    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/rD1F24V1ORDI' 258368
)
    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/rD1F24V2ORDI' 258368
)
    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/rD1F24V3ORDI' 258368
)
    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/rD1F24V4ORDI' 258368
)
    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/rD1F25V1ORDI' 258368
)
    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/rD1F25V2ORDI' 258368
)
    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/rD1F25V3ORDI' 258368
)
    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/rD1F25V4ORDI' 258368
)
    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/rD1F26V1ORDI' 258368
)
    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/rD1F26V2ORDI' 258368
)
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/rD1F26V3ORDI' 258368
)
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/rD1F26V4ORDI' 258368
)
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/rD1F27V1ORDI' 258368
)
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/rD1F27V2ORDI' 258368
)
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/rD1F27V3ORDI' 258368
)
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/rD1F27V4ORDI' 258368
)
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/rD1F28V1ORDI' 258368
)
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/rD1F28V2ORDI' 258368
)
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/rD1F28V3ORDI' 258368
)
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/rD1F28V4ORDI' 258368
)
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/rD1F29V1ORDI' 258368
)
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/rD1F29V2ORDI' 258368
)
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/rD1F29V3ORDI' 258368
)
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/rD1F29V4ORDI' 258368
)
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/rD1F30V1ORDI' 258368
)

```

```

    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/rD1F30V2ORDI' 258368
)
    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/rD1F30V3ORDI' 258368
)
    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/rD1F30V4ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_121 of D1
drop tablespace is_order_121;
create regular tablespace is_order_121 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V1ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_122 of D1
drop tablespace is_order_122;
create regular tablespace is_order_122 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V2ORDI' 258368
)
    extentsize 256
    prefetchsize 4096

```

```

    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_123 of D1
drop tablespace is_order_123;
create regular tablespace is_order_123 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V3ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_124 of D1
drop tablespace is_order_124;
create regular tablespace is_order_124 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V4ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_125 of D1
drop tablespace is_order_125;
create regular tablespace is_order_125 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V1ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_126 of D1
drop tablespace is_order_126;
create regular tablespace is_order_126 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V2ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_127 of D1
drop tablespace is_order_127;
create regular tablespace is_order_127 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V3ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_128 of D1
drop tablespace is_order_128;
create regular tablespace is_order_128 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V4ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_129 of D1
drop tablespace is_order_129;
create regular tablespace is_order_129 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V1ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_130 of D1
drop tablespace is_order_130;
create regular tablespace is_order_130 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V2ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_131 of D1
drop tablespace is_order_131;
create regular tablespace is_order_131 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V3ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_132 of D1
drop tablespace is_order_132;
create regular tablespace is_order_132 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V4ORDI' 258368
)
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_133 of D1
drop tablespace is_order_133;

```

```

create regular tablespace is_order_133 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V1ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_134 of D1
drop tablespace is_order_134;
create regular tablespace is_order_134 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_135 of D1
drop tablespace is_order_135;
create regular tablespace is_order_135 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_136 of D1
drop tablespace is_order_136;
create regular tablespace is_order_136 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V4ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_137 of D1
drop tablespace is_order_137;
create regular tablespace is_order_137 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V1ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_138 of D1
drop tablespace is_order_138;
create regular tablespace is_order_138 pagesize 8K
managed by database

```

```

using
(
    device '/dev/rD1F35V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_139 of D1
drop tablespace is_order_139;
create regular tablespace is_order_139 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_140 of D1
drop tablespace is_order_140;
create regular tablespace is_order_140 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V4ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_141 of D1
drop tablespace is_order_141;
create regular tablespace is_order_141 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V1ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_142 of D1
drop tablespace is_order_142;
create regular tablespace is_order_142 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_143 of D1
drop tablespace is_order_143;
create regular tablespace is_order_143 pagesize 8K
managed by database
using
(

```

```

    device '/dev/rD1F36V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_144 of D1
drop tablespace is_order_144;
create regular tablespace is_order_144 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V4ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_145 of D1
drop tablespace is_order_145;
create regular tablespace is_order_145 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V1ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_146 of D1
drop tablespace is_order_146;
create regular tablespace is_order_146 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_147 of D1
drop tablespace is_order_147;
create regular tablespace is_order_147 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_148 of D1
drop tablespace is_order_148;
create regular tablespace is_order_148 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V4ORDI' 258368
)

```

```

    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_149 of D1
drop tablespace is_order_149;
create regular tablespace is_order_149 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V1ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_150 of D1
drop tablespace is_order_150;
create regular tablespace is_order_150 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_151 of D1
drop tablespace is_order_151;
create regular tablespace is_order_151 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_152 of D1
drop tablespace is_order_152;
create regular tablespace is_order_152 pagesize 8K
managed by database
using
(
    device '/dev/rD1F38V4ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_153 of D1
drop tablespace is_order_153;
create regular tablespace is_order_153 pagesize 8K
managed by database
using
(
    device '/dev/rD1F39V1ORDI' 258368
)
extentsize 256
prefetchsize 4096

```

```

    bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_154 of D1
drop tablespace is_order_154;
create regular tablespace is_order_154 pagesize 8K
managed by database
using
(
    device '/dev/rD1F39V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_155 of D1
drop tablespace is_order_155;
create regular tablespace is_order_155 pagesize 8K
managed by database
using
(
    device '/dev/rD1F39V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_156 of D1
drop tablespace is_order_156;
create regular tablespace is_order_156 pagesize 8K
managed by database
using
(
    device '/dev/rD1F39V4ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_157 of D1
drop tablespace is_order_157;
create regular tablespace is_order_157 pagesize 8K
managed by database
using
(
    device '/dev/rD1F40V1ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_158 of D1
drop tablespace is_order_158;
create regular tablespace is_order_158 pagesize 8K
managed by database
using
(
    device '/dev/rD1F40V2ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_159 of D1
drop tablespace is_order_159;
create regular tablespace is_order_159 pagesize 8K
managed by database
using
(
    device '/dev/rD1F40V3ORDI' 258368
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for is_order_160 of D1
drop tablespace is_order_160;
create regular tablespace is_order_160 pagesize 8K
managed by database
using
(
    device '/dev/rD1F40V4ORDI' 258368
)
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' 10081536
)
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' 10081536
)
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' 10081536
)
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' 10081536
)
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/rD1F02V1CST' 10081536
)
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/rD1F02V2CST' 10081536
)
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/rD1F02V3CST' 10081536
)
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/rD1F02V4CST' 10081536
)
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/rD1F03V1CST' 10081536
)
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/rD1F03V2CST' 10081536
)
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/rD1F03V3CST' 10081536
)
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/rD1F03V4CST' 10081536
)
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/rD1F04V1CST' 10081536
)
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/rD1F04V2CST' 10081536
)
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/rD1F04V3CST' 10081536
)
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/rD1F04V4CST' 10081536
)
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/rD1F05V1CST' 10081536
)
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/rD1F05V2CST' 10081536
)
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/rD1F05V3CST' 10081536
)
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/rD1F05V4CST' 10081536
)
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/rD1F06V1CST' 10081536
)
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/rD1F06V2CST' 10081536
)
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/rD1F06V3CST' 10081536
)
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/rD1F06V4CST' 10081536
)
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/rD1F07V1CST' 10081536
)
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/rD1F07V2CST' 10081536
)
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/rD1F07V3CST' 10081536
)
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/rD1F07V4CST' 10081536
)
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/rD1F08V1CST' 10081536
)
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/rD1F08V2CST' 10081536
)
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/rD1F08V3CST' 10081536
)
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/rD1F08V4CST' 10081536
)
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/rD1F09V1CST' 10081536
)
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/rD1F09V2CST' 10081536
)
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/rD1F09V3CST' 10081536
)
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/rD1F09V4CST' 10081536
)
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/rD1F10V1CST' 10081536
)

```

```

        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/rD1F10V2CST' 10081536
)
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/rD1F10V3CST' 10081536
)
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/rD1F10V4CST' 10081536
)
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/rD1F11V1CST' 10081536
)
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/rD1F11V2CST' 10081536
)
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/rD1F11V3CST' 10081536
)
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/rD1F11V4CST' 10081536
)
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/rD1F12V1CST' 10081536
)
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/rD1F12V2CST' 10081536
)
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/rD1F12V3CST' 10081536
)
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/rD1F12V4CST' 10081536
)
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/rD1F13V1CST' 10081536
)
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/rD1F13V2CST' 10081536
)
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/rD1F13V3CST' 10081536
)
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/rD1F13V4CST' 10081536
)
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/rD1F14V1CST' 10081536
)
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/rD1F14V2CST' 10081536
)

```



```

    )
    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/rD1F14V3CST' 10081536
)
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/rD1F14V4CST' 10081536
)
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/rD1F15V1CST' 10081536
)
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/rD1F15V2CST' 10081536
)
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/rD1F15V3CST' 10081536
)
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/rD1F15V4CST' 10081536
)
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/rD1F16V1CST' 10081536
)
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/rD1F16V2CST' 10081536
)
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/rD1F16V3CST' 10081536
)
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/rD1F16V4CST' 10081536
)
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/rD1F17V1CST' 10081536
)
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/rD1F17V2CST' 10081536
)
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/rD1F17V3CST' 10081536
)
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/rD1F17V4CST' 10081536
)
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/rD1F18V1CST' 10081536
)
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/rD1F18V2CST' 10081536
)
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/rD1F18V3CST' 10081536
    )
    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/rD1F18V4CST' 10081536
)
    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/rD1F19V1CST' 10081536
)
    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/rD1F19V2CST' 10081536
)
    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/rD1F19V3CST' 10081536
)
    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/rD1F19V4CST' 10081536
)
    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/rD1F20V1CST' 10081536
)
    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/rD1F20V2CST' 10081536
)
    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/rD1F20V3CST' 10081536
)
    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/rD1F20V4CST' 10081536
)
    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/rD1F21V1CST' 10081536
)
    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/rD1F21V2CST' 10081536
)
    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/rD1F21V3CST' 10081536
)
    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/rD1F21V4CST' 10081536
)
    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/rD1F22V1CST' 10081536
)
    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/rD1F22V2CST' 10081536
)
    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/rD1F22V3CST' 10081536
)
    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/rD1F22V4CST' 10081536
)
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/rD1F23V1CST' 10081536
)
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/rD1F23V2CST' 10081536
)
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/rD1F23V3CST' 10081536
)
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/rD1F23V4CST' 10081536
)
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/rD1F24V1CST' 10081536
)
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/rD1F24V2CST' 10081536
)
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/rD1F24V3CST' 10081536
)
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/rD1F24V4CST' 10081536
)
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/rD1F25V1CST' 10081536
)
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/rD1F25V2CST' 10081536
)
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/rD1F25V3CST' 10081536
)

```

```

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/rD1F25V4CST' 10081536
)
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/rD1F26V1CST' 10081536
)
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/rD1F26V2CST' 10081536
)
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/rD1F26V3CST' 10081536
)
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/rD1F26V4CST' 10081536
)
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/rD1F27V1CST' 10081536
)
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/rD1F27V2CST' 10081536
)
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/rD1F27V3CST' 10081536
)
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/rD1F27V4CST' 10081536
)
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/rD1F28V1CST' 10081536
)
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/rD1F28V2CST' 10081536
)
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/rD1F28V3CST' 10081536
)
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/rD1F28V4CST' 10081536
)
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/rD1F29V1CST' 10081536
)
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/rD1F29V2CST' 10081536
)
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/rD1F29V3CST' 10081536
)
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/rD1F29V4CST' 10081536
)

```

```

)
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/rD1F30V1CST' 10081536
)
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/rD1F30V2CST' 10081536
)
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/rD1F30V3CST' 10081536
)
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/rD1F30V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_121 of D1
drop tablespace ts_customer_121;
create regular tablespace ts_customer_121 pagesize 4K
managed by database
using
(
    device '/dev/rD1F31V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_122 of D1
drop tablespace ts_customer_122;
create regular tablespace ts_customer_122 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F31V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_123 of D1
drop tablespace ts_customer_123;
create regular tablespace ts_customer_123 pagesize 4K
managed by database
using
(
    device '/dev/rD1F31V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_124 of D1
drop tablespace ts_customer_124;
create regular tablespace ts_customer_124 pagesize 4K
managed by database
using
(
    device '/dev/rD1F31V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_125 of D1
drop tablespace ts_customer_125;
create regular tablespace ts_customer_125 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_126 of D1
drop tablespace ts_customer_126;
create regular tablespace ts_customer_126 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_127 of D1
drop tablespace ts_customer_127;
create regular tablespace ts_customer_127 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V3CST' 10081536
)
extentsize 256
prefetchsize 4096;

```

```

commit;
-- now creating TS for ts_customer_128 of D1
drop tablespace ts_customer_128;
create regular tablespace ts_customer_128 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_129 of D1
drop tablespace ts_customer_129;
create regular tablespace ts_customer_129 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_130 of D1
drop tablespace ts_customer_130;
create regular tablespace ts_customer_130 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_131 of D1
drop tablespace ts_customer_131;
create regular tablespace ts_customer_131 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_132 of D1
drop tablespace ts_customer_132;
create regular tablespace ts_customer_132 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_133 of D1
drop tablespace ts_customer_133;
create regular tablespace ts_customer_133 pagesize 4K
managed by database
using
(

```

```

    device '/dev/rD1F34V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_134 of D1
drop tablespace ts_customer_134;
create regular tablespace ts_customer_134 pagesize 4K
managed by database
using
(
    device '/dev/rD1F34V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_135 of D1
drop tablespace ts_customer_135;
create regular tablespace ts_customer_135 pagesize 4K
managed by database
using
(
    device '/dev/rD1F34V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_136 of D1
drop tablespace ts_customer_136;
create regular tablespace ts_customer_136 pagesize 4K
managed by database
using
(
    device '/dev/rD1F34V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_137 of D1
drop tablespace ts_customer_137;
create regular tablespace ts_customer_137 pagesize 4K
managed by database
using
(
    device '/dev/rD1F35V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_138 of D1
drop tablespace ts_customer_138;
create regular tablespace ts_customer_138 pagesize 4K
managed by database
using
(
    device '/dev/rD1F35V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_139 of D1
drop tablespace ts_customer_139;

```

```

create regular tablespace ts_customer_139 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F35V3CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_140 of D1
drop tablespace ts_customer_140;
create regular tablespace ts_customer_140 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F35V4CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_141 of D1
drop tablespace ts_customer_141;
create regular tablespace ts_customer_141 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F36V1CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_142 of D1
drop tablespace ts_customer_142;
create regular tablespace ts_customer_142 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F36V2CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_143 of D1
drop tablespace ts_customer_143;
create regular tablespace ts_customer_143 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F36V3CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_144 of D1
drop tablespace ts_customer_144;
create regular tablespace ts_customer_144 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F36V4CST' 10081536
  )
  extentsize 256

```

```

  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_145 of D1
drop tablespace ts_customer_145;
create regular tablespace ts_customer_145 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F37V1CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_146 of D1
drop tablespace ts_customer_146;
create regular tablespace ts_customer_146 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F37V2CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_147 of D1
drop tablespace ts_customer_147;
create regular tablespace ts_customer_147 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F37V3CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_148 of D1
drop tablespace ts_customer_148;
create regular tablespace ts_customer_148 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F37V4CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_149 of D1
drop tablespace ts_customer_149;
create regular tablespace ts_customer_149 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F38V1CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_150 of D1
drop tablespace ts_customer_150;
create regular tablespace ts_customer_150 pagesize 4K
  managed by database
  using

```

```

  (
    device '/dev/rD1F38V2CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_151 of D1
drop tablespace ts_customer_151;
create regular tablespace ts_customer_151 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F38V3CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_152 of D1
drop tablespace ts_customer_152;
create regular tablespace ts_customer_152 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F38V4CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_153 of D1
drop tablespace ts_customer_153;
create regular tablespace ts_customer_153 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F39V1CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_154 of D1
drop tablespace ts_customer_154;
create regular tablespace ts_customer_154 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F39V2CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_155 of D1
drop tablespace ts_customer_155;
create regular tablespace ts_customer_155 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F39V3CST' 10081536
  )
  extentsize 256
  prefetchsize 4096;
commit;
-- now creating TS for ts_customer_156 of D1

```

```

drop tablespace ts_customer_156;
create regular tablespace ts_customer_156 pagesize 4K
managed by database
using
(
    device '/dev/rD1F39V4CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_157 of D1
drop tablespace ts_customer_157;
create regular tablespace ts_customer_157 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V1CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_158 of D1
drop tablespace ts_customer_158;
create regular tablespace ts_customer_158 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V2CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_159 of D1
drop tablespace ts_customer_159;
create regular tablespace ts_customer_159 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V3CST' 10081536
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_customer_160 of D1
drop tablespace ts_customer_160;
create regular tablespace ts_customer_160 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V4CST' 10081536
)
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/rD1F01V1DIST' 1024,
    device '/dev/rD1F01V2DIST' 1024,
    device '/dev/rD1F01V3DIST' 1024,
    device '/dev/rD1F01V4DIST' 1024,
    device '/dev/rD1F02V1DIST' 1024,
    device '/dev/rD1F02V2DIST' 1024,
    device '/dev/rD1F02V3DIST' 1024,
    device '/dev/rD1F02V4DIST' 1024,
    device '/dev/rD1F03V1DIST' 1024,
    device '/dev/rD1F03V2DIST' 1024,
    device '/dev/rD1F03V3DIST' 1024,
    device '/dev/rD1F03V4DIST' 1024,
    device '/dev/rD1F04V1DIST' 1024,
    device '/dev/rD1F04V2DIST' 1024,
    device '/dev/rD1F04V3DIST' 1024,
    device '/dev/rD1F04V4DIST' 1024,
    device '/dev/rD1F05V1DIST' 1024,
    device '/dev/rD1F05V2DIST' 1024,
    device '/dev/rD1F05V3DIST' 1024,
    device '/dev/rD1F05V4DIST' 1024
)
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/rD1F06V1DIST' 1024,
    device '/dev/rD1F06V2DIST' 1024,
    device '/dev/rD1F06V3DIST' 1024,
    device '/dev/rD1F06V4DIST' 1024,
    device '/dev/rD1F07V1DIST' 1024,
    device '/dev/rD1F07V2DIST' 1024,
    device '/dev/rD1F07V3DIST' 1024,
    device '/dev/rD1F07V4DIST' 1024,
    device '/dev/rD1F08V1DIST' 1024,
    device '/dev/rD1F08V2DIST' 1024,
    device '/dev/rD1F08V3DIST' 1024,
    device '/dev/rD1F08V4DIST' 1024,
    device '/dev/rD1F09V1DIST' 1024,
    device '/dev/rD1F09V2DIST' 1024,
    device '/dev/rD1F09V3DIST' 1024,
    device '/dev/rD1F09V4DIST' 1024,
    device '/dev/rD1F10V1DIST' 1024,
    device '/dev/rD1F10V2DIST' 1024,
    device '/dev/rD1F10V3DIST' 1024,
    device '/dev/rD1F10V4DIST' 1024
)
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/rD1F11V1DIST' 1024,
    device '/dev/rD1F11V2DIST' 1024,
    device '/dev/rD1F11V3DIST' 1024,
    device '/dev/rD1F11V4DIST' 1024,
    device '/dev/rD1F12V1DIST' 1024,
    device '/dev/rD1F12V2DIST' 1024,
    device '/dev/rD1F12V3DIST' 1024,
    device '/dev/rD1F12V4DIST' 1024,
    device '/dev/rD1F13V1DIST' 1024,
    device '/dev/rD1F13V2DIST' 1024,
    device '/dev/rD1F13V3DIST' 1024,
    device '/dev/rD1F13V4DIST' 1024,
    device '/dev/rD1F14V1DIST' 1024,
    device '/dev/rD1F14V2DIST' 1024,
    device '/dev/rD1F14V3DIST' 1024,
    device '/dev/rD1F14V4DIST' 1024,
    device '/dev/rD1F15V1DIST' 1024,
    device '/dev/rD1F15V2DIST' 1024,
    device '/dev/rD1F15V3DIST' 1024,
    device '/dev/rD1F15V4DIST' 1024
)
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/rD1F16V1DIST' 1024,
    device '/dev/rD1F16V2DIST' 1024,
    device '/dev/rD1F16V3DIST' 1024,
    device '/dev/rD1F16V4DIST' 1024,
    device '/dev/rD1F17V1DIST' 1024,
    device '/dev/rD1F17V2DIST' 1024,
    device '/dev/rD1F17V3DIST' 1024,
    device '/dev/rD1F17V4DIST' 1024,
    device '/dev/rD1F18V1DIST' 1024,
    device '/dev/rD1F18V2DIST' 1024,
    device '/dev/rD1F18V3DIST' 1024,
    device '/dev/rD1F18V4DIST' 1024,
    device '/dev/rD1F19V1DIST' 1024,
    device '/dev/rD1F19V2DIST' 1024,
    device '/dev/rD1F19V3DIST' 1024,
    device '/dev/rD1F19V4DIST' 1024,
    device '/dev/rD1F20V1DIST' 1024,
    device '/dev/rD1F20V2DIST' 1024,
    device '/dev/rD1F20V3DIST' 1024,
    device '/dev/rD1F20V4DIST' 1024
)
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/rD1F21V1DIST' 1024,
device '/dev/rD1F21V2DIST' 1024,
device '/dev/rD1F21V3DIST' 1024,
device '/dev/rD1F21V4DIST' 1024,
device '/dev/rD1F22V1DIST' 1024,
device '/dev/rD1F22V2DIST' 1024,
device '/dev/rD1F22V3DIST' 1024,
device '/dev/rD1F22V4DIST' 1024,
device '/dev/rD1F23V1DIST' 1024,
device '/dev/rD1F23V2DIST' 1024,
device '/dev/rD1F23V3DIST' 1024,
device '/dev/rD1F23V4DIST' 1024,
device '/dev/rD1F24V1DIST' 1024,
device '/dev/rD1F24V2DIST' 1024,
device '/dev/rD1F24V3DIST' 1024,
device '/dev/rD1F24V4DIST' 1024,
device '/dev/rD1F25V1DIST' 1024,
device '/dev/rD1F25V2DIST' 1024,
device '/dev/rD1F25V3DIST' 1024,
device '/dev/rD1F25V4DIST' 1024
)
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/rD1F26V1DIST' 1024,
device '/dev/rD1F26V2DIST' 1024,
device '/dev/rD1F26V3DIST' 1024,
device '/dev/rD1F26V4DIST' 1024,
device '/dev/rD1F27V1DIST' 1024,
device '/dev/rD1F27V2DIST' 1024,
device '/dev/rD1F27V3DIST' 1024,
device '/dev/rD1F27V4DIST' 1024,
device '/dev/rD1F28V1DIST' 1024,
device '/dev/rD1F28V2DIST' 1024,
device '/dev/rD1F28V3DIST' 1024,
device '/dev/rD1F28V4DIST' 1024,
device '/dev/rD1F29V1DIST' 1024,
device '/dev/rD1F29V2DIST' 1024,
device '/dev/rD1F29V3DIST' 1024,
device '/dev/rD1F29V4DIST' 1024,
device '/dev/rD1F30V1DIST' 1024,
device '/dev/rD1F30V2DIST' 1024,
device '/dev/rD1F30V3DIST' 1024,
device '/dev/rD1F30V4DIST' 1024
)
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/rD1F31V1DIST' 1024,
device '/dev/rD1F31V2DIST' 1024,

```

```

device '/dev/rD1F31V3DIST' 1024,
device '/dev/rD1F31V4DIST' 1024,
device '/dev/rD1F32V1DIST' 1024,
device '/dev/rD1F32V2DIST' 1024,
device '/dev/rD1F32V3DIST' 1024,
device '/dev/rD1F32V4DIST' 1024,
device '/dev/rD1F33V1DIST' 1024,
device '/dev/rD1F33V2DIST' 1024,
device '/dev/rD1F33V3DIST' 1024,
device '/dev/rD1F33V4DIST' 1024,
device '/dev/rD1F34V1DIST' 1024,
device '/dev/rD1F34V2DIST' 1024,
device '/dev/rD1F34V3DIST' 1024,
device '/dev/rD1F34V4DIST' 1024,
device '/dev/rD1F35V1DIST' 1024,
device '/dev/rD1F35V2DIST' 1024,
device '/dev/rD1F35V3DIST' 1024,
device '/dev/rD1F35V4DIST' 1024
)
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/rD1F36V1DIST' 1024,
device '/dev/rD1F36V2DIST' 1024,
device '/dev/rD1F36V3DIST' 1024,
device '/dev/rD1F36V4DIST' 1024,
device '/dev/rD1F37V1DIST' 1024,
device '/dev/rD1F37V2DIST' 1024,
device '/dev/rD1F37V3DIST' 1024,
device '/dev/rD1F37V4DIST' 1024,
device '/dev/rD1F38V1DIST' 1024,
device '/dev/rD1F38V2DIST' 1024,
device '/dev/rD1F38V3DIST' 1024,
device '/dev/rD1F38V4DIST' 1024,
device '/dev/rD1F39V1DIST' 1024,
device '/dev/rD1F39V2DIST' 1024,
device '/dev/rD1F39V3DIST' 1024,
device '/dev/rD1F39V4DIST' 1024,
device '/dev/rD1F40V1DIST' 1024,
device '/dev/rD1F40V2DIST' 1024,
device '/dev/rD1F40V3DIST' 1024,
device '/dev/rD1F40V4DIST' 1024
)
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' 284416,
device '/dev/rD1F01V2HIST' 284416,
device '/dev/rD1F01V3HIST' 284416,
device '/dev/rD1F01V4HIST' 284416,
device '/dev/rD1F02V1HIST' 284416,
device '/dev/rD1F02V2HIST' 284416,
device '/dev/rD1F02V3HIST' 284416,
device '/dev/rD1F02V4HIST' 284416,
device '/dev/rD1F03V1HIST' 284416,
device '/dev/rD1F03V2HIST' 284416
)
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/rD1F03V3HIST' 284416,
device '/dev/rD1F03V4HIST' 284416,
device '/dev/rD1F04V1HIST' 284416,
device '/dev/rD1F04V2HIST' 284416,
device '/dev/rD1F04V3HIST' 284416,
device '/dev/rD1F04V4HIST' 284416,
device '/dev/rD1F05V1HIST' 284416,
device '/dev/rD1F05V2HIST' 284416,
device '/dev/rD1F05V3HIST' 284416,
device '/dev/rD1F05V4HIST' 284416
)
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/rD1F06V1HIST' 284416,
device '/dev/rD1F06V2HIST' 284416,
device '/dev/rD1F06V3HIST' 284416,
device '/dev/rD1F06V4HIST' 284416,
device '/dev/rD1F07V1HIST' 284416,
device '/dev/rD1F07V2HIST' 284416,
device '/dev/rD1F07V3HIST' 284416,
device '/dev/rD1F07V4HIST' 284416,
device '/dev/rD1F08V1HIST' 284416,
device '/dev/rD1F08V2HIST' 284416
)
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/rD1F08V3HIST' 284416,
    device '/dev/rD1F08V4HIST' 284416,
    device '/dev/rD1F09V1HIST' 284416,
    device '/dev/rD1F09V2HIST' 284416,
    device '/dev/rD1F09V3HIST' 284416,
    device '/dev/rD1F09V4HIST' 284416,
    device '/dev/rD1F10V1HIST' 284416,
    device '/dev/rD1F10V2HIST' 284416,
    device '/dev/rD1F10V3HIST' 284416,
    device '/dev/rD1F10V4HIST' 284416
)
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/rD1F11V1HIST' 284416,
    device '/dev/rD1F11V2HIST' 284416,
    device '/dev/rD1F11V3HIST' 284416,
    device '/dev/rD1F11V4HIST' 284416,
    device '/dev/rD1F12V1HIST' 284416,
    device '/dev/rD1F12V2HIST' 284416,
    device '/dev/rD1F12V3HIST' 284416,
    device '/dev/rD1F12V4HIST' 284416,
    device '/dev/rD1F13V1HIST' 284416,
    device '/dev/rD1F13V2HIST' 284416
)
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/rD1F13V3HIST' 284416,
    device '/dev/rD1F13V4HIST' 284416,
    device '/dev/rD1F14V1HIST' 284416,
    device '/dev/rD1F14V2HIST' 284416,
    device '/dev/rD1F14V3HIST' 284416,
    device '/dev/rD1F14V4HIST' 284416,
    device '/dev/rD1F15V1HIST' 284416,
    device '/dev/rD1F15V2HIST' 284416,
    device '/dev/rD1F15V3HIST' 284416,
    device '/dev/rD1F15V4HIST' 284416
)
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/rD1F16V1HIST' 284416,
    device '/dev/rD1F16V2HIST' 284416,
    device '/dev/rD1F16V3HIST' 284416,
    device '/dev/rD1F16V4HIST' 284416,
    device '/dev/rD1F17V1HIST' 284416,
    device '/dev/rD1F17V2HIST' 284416,
    device '/dev/rD1F17V3HIST' 284416,
    device '/dev/rD1F17V4HIST' 284416,
    device '/dev/rD1F18V1HIST' 284416,
    device '/dev/rD1F18V2HIST' 284416
)
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/rD1F18V3HIST' 284416,
    device '/dev/rD1F18V4HIST' 284416,
    device '/dev/rD1F19V1HIST' 284416,
    device '/dev/rD1F19V2HIST' 284416,
    device '/dev/rD1F19V3HIST' 284416,
    device '/dev/rD1F19V4HIST' 284416,
    device '/dev/rD1F20V1HIST' 284416,
    device '/dev/rD1F20V2HIST' 284416,
    device '/dev/rD1F20V3HIST' 284416,
    device '/dev/rD1F20V4HIST' 284416
)
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/rD1F21V1HIST' 284416,
    device '/dev/rD1F21V2HIST' 284416,
    device '/dev/rD1F21V3HIST' 284416,
    device '/dev/rD1F21V4HIST' 284416,
    device '/dev/rD1F22V1HIST' 284416,
    device '/dev/rD1F22V2HIST' 284416,
    device '/dev/rD1F22V3HIST' 284416,
    device '/dev/rD1F22V4HIST' 284416,
    device '/dev/rD1F23V1HIST' 284416,
    device '/dev/rD1F23V2HIST' 284416
)
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/rD1F23V3HIST' 284416,
    device '/dev/rD1F23V4HIST' 284416,
    device '/dev/rD1F24V1HIST' 284416,
    device '/dev/rD1F24V2HIST' 284416,
    device '/dev/rD1F24V3HIST' 284416,
    device '/dev/rD1F24V4HIST' 284416,
    device '/dev/rD1F25V1HIST' 284416,
    device '/dev/rD1F25V2HIST' 284416,
    device '/dev/rD1F25V3HIST' 284416,
    device '/dev/rD1F25V4HIST' 284416
)
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/rD1F26V1HIST' 284416,
    device '/dev/rD1F26V2HIST' 284416,
    device '/dev/rD1F26V3HIST' 284416,
    device '/dev/rD1F26V4HIST' 284416,
    device '/dev/rD1F27V1HIST' 284416,
    device '/dev/rD1F27V2HIST' 284416,
    device '/dev/rD1F27V3HIST' 284416,
    device '/dev/rD1F27V4HIST' 284416,
    device '/dev/rD1F28V1HIST' 284416,
    device '/dev/rD1F28V2HIST' 284416
)
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/rD1F28V3HIST' 284416,
    device '/dev/rD1F28V4HIST' 284416,
    device '/dev/rD1F29V1HIST' 284416,
    device '/dev/rD1F29V2HIST' 284416,
    device '/dev/rD1F29V3HIST' 284416,
    device '/dev/rD1F29V4HIST' 284416,
    device '/dev/rD1F30V1HIST' 284416,
    device '/dev/rD1F30V2HIST' 284416,
    device '/dev/rD1F30V3HIST' 284416,
    device '/dev/rD1F30V4HIST' 284416
)
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/rD1F31V1HIST' 284416,
    device '/dev/rD1F31V2HIST' 284416,
    device '/dev/rD1F31V3HIST' 284416,
    device '/dev/rD1F31V4HIST' 284416,
    device '/dev/rD1F32V1HIST' 284416,
    device '/dev/rD1F32V2HIST' 284416,
    device '/dev/rD1F32V3HIST' 284416,
    device '/dev/rD1F32V4HIST' 284416,
    device '/dev/rD1F33V1HIST' 284416,
    device '/dev/rD1F33V2HIST' 284416
)
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/rD1F33V3HIST' 284416,
    device '/dev/rD1F33V4HIST' 284416,
    device '/dev/rD1F34V1HIST' 284416,
    device '/dev/rD1F34V2HIST' 284416,
    device '/dev/rD1F34V3HIST' 284416,
    device '/dev/rD1F34V4HIST' 284416,
    device '/dev/rD1F35V1HIST' 284416,
    device '/dev/rD1F35V2HIST' 284416,
    device '/dev/rD1F35V3HIST' 284416,
    device '/dev/rD1F35V4HIST' 284416
)
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/rD1F36V1HIST' 284416,
    device '/dev/rD1F36V2HIST' 284416,
    device '/dev/rD1F36V3HIST' 284416,
    device '/dev/rD1F36V4HIST' 284416,
    device '/dev/rD1F37V1HIST' 284416,
    device '/dev/rD1F37V2HIST' 284416,
    device '/dev/rD1F37V3HIST' 284416,
    device '/dev/rD1F37V4HIST' 284416,
    device '/dev/rD1F38V1HIST' 284416,
    device '/dev/rD1F38V2HIST' 284416
)
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/rD1F38V3HIST' 284416,
    device '/dev/rD1F38V4HIST' 284416,
    device '/dev/rD1F39V1HIST' 284416,
    device '/dev/rD1F39V2HIST' 284416,
    device '/dev/rD1F39V3HIST' 284416,
    device '/dev/rD1F39V4HIST' 284416,
    device '/dev/rD1F40V1HIST' 284416,
    device '/dev/rD1F40V2HIST' 284416,
    device '/dev/rD1F40V3HIST' 284416,
    device '/dev/rD1F40V4HIST' 284416
)
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 4K
managed by database
using
(
    device '/dev/rD1F01V1ITEM' 128,
    device '/dev/rD1F01V2ITEM' 128,
    device '/dev/rD1F01V3ITEM' 128,
    device '/dev/rD1F01V4ITEM' 128,
    device '/dev/rD1F02V1ITEM' 128,
    device '/dev/rD1F02V2ITEM' 128,
    device '/dev/rD1F02V3ITEM' 128,
    device '/dev/rD1F02V4ITEM' 128,
    device '/dev/rD1F03V1ITEM' 128,
    device '/dev/rD1F03V2ITEM' 128,
    device '/dev/rD1F03V3ITEM' 128,
    device '/dev/rD1F03V4ITEM' 128,
    device '/dev/rD1F04V1ITEM' 128,
    device '/dev/rD1F04V2ITEM' 128,
    device '/dev/rD1F04V3ITEM' 128,
    device '/dev/rD1F04V4ITEM' 128,
    device '/dev/rD1F05V1ITEM' 128,
    device '/dev/rD1F05V2ITEM' 128,
    device '/dev/rD1F05V3ITEM' 128,
    device '/dev/rD1F05V4ITEM' 128,
    device '/dev/rD1F06V1ITEM' 128,
    device '/dev/rD1F06V2ITEM' 128,
    device '/dev/rD1F06V3ITEM' 128,
    device '/dev/rD1F06V4ITEM' 128,
    device '/dev/rD1F07V1ITEM' 128,
    device '/dev/rD1F07V2ITEM' 128,
    device '/dev/rD1F07V3ITEM' 128,
    device '/dev/rD1F07V4ITEM' 128,

```

```

device '/dev/rD1F08V1ITEM' 128,
device '/dev/rD1F08V2ITEM' 128,
device '/dev/rD1F08V3ITEM' 128,
device '/dev/rD1F08V4ITEM' 128,
device '/dev/rD1F09V1ITEM' 128,
device '/dev/rD1F09V2ITEM' 128,
device '/dev/rD1F09V3ITEM' 128,
device '/dev/rD1F09V4ITEM' 128,
device '/dev/rD1F10V1ITEM' 128,
device '/dev/rD1F10V2ITEM' 128,
device '/dev/rD1F10V3ITEM' 128,
device '/dev/rD1F10V4ITEM' 128,
device '/dev/rD1F11V1ITEM' 128,
device '/dev/rD1F11V2ITEM' 128,
device '/dev/rD1F11V3ITEM' 128,
device '/dev/rD1F11V4ITEM' 128,
device '/dev/rD1F12V1ITEM' 128,
device '/dev/rD1F12V2ITEM' 128,
device '/dev/rD1F12V3ITEM' 128,
device '/dev/rD1F12V4ITEM' 128,
device '/dev/rD1F13V1ITEM' 128,
device '/dev/rD1F13V2ITEM' 128,
device '/dev/rD1F13V3ITEM' 128,
device '/dev/rD1F13V4ITEM' 128,
device '/dev/rD1F14V1ITEM' 128,
device '/dev/rD1F14V2ITEM' 128,
device '/dev/rD1F14V3ITEM' 128,
device '/dev/rD1F14V4ITEM' 128,
device '/dev/rD1F15V1ITEM' 128,
device '/dev/rD1F15V2ITEM' 128,
device '/dev/rD1F15V3ITEM' 128,
device '/dev/rD1F15V4ITEM' 128,
device '/dev/rD1F16V1ITEM' 128,
device '/dev/rD1F16V2ITEM' 128,
device '/dev/rD1F16V3ITEM' 128,
device '/dev/rD1F16V4ITEM' 128,
device '/dev/rD1F17V1ITEM' 128,
device '/dev/rD1F17V2ITEM' 128,
device '/dev/rD1F17V3ITEM' 128,
device '/dev/rD1F17V4ITEM' 128,
device '/dev/rD1F18V1ITEM' 128,
device '/dev/rD1F18V2ITEM' 128,
device '/dev/rD1F18V3ITEM' 128,
device '/dev/rD1F18V4ITEM' 128,
device '/dev/rD1F19V1ITEM' 128,
device '/dev/rD1F19V2ITEM' 128,
device '/dev/rD1F19V3ITEM' 128,
device '/dev/rD1F19V4ITEM' 128,
device '/dev/rD1F20V1ITEM' 128,
device '/dev/rD1F20V2ITEM' 128,
device '/dev/rD1F20V3ITEM' 128,
device '/dev/rD1F20V4ITEM' 128,
device '/dev/rD1F21V1ITEM' 128,
device '/dev/rD1F21V2ITEM' 128,
device '/dev/rD1F21V3ITEM' 128,
device '/dev/rD1F21V4ITEM' 128,
device '/dev/rD1F22V1ITEM' 128,
device '/dev/rD1F22V2ITEM' 128,
device '/dev/rD1F22V3ITEM' 128,
device '/dev/rD1F22V4ITEM' 128,
device '/dev/rD1F23V1ITEM' 128,
device '/dev/rD1F23V2ITEM' 128,

```

```

device '/dev/rD1F23V3ITEM' 128,
device '/dev/rD1F23V4ITEM' 128,
device '/dev/rD1F24V1ITEM' 128,
device '/dev/rD1F24V2ITEM' 128,
device '/dev/rD1F24V3ITEM' 128,
device '/dev/rD1F24V4ITEM' 128,
device '/dev/rD1F25V1ITEM' 128,
device '/dev/rD1F25V2ITEM' 128,
device '/dev/rD1F25V3ITEM' 128,
device '/dev/rD1F25V4ITEM' 128,
device '/dev/rD1F26V1ITEM' 128,
device '/dev/rD1F26V2ITEM' 128,
device '/dev/rD1F26V3ITEM' 128,
device '/dev/rD1F26V4ITEM' 128,
device '/dev/rD1F27V1ITEM' 128,
device '/dev/rD1F27V2ITEM' 128,
device '/dev/rD1F27V3ITEM' 128,
device '/dev/rD1F27V4ITEM' 128,
device '/dev/rD1F28V1ITEM' 128,
device '/dev/rD1F28V2ITEM' 128,
device '/dev/rD1F28V3ITEM' 128,
device '/dev/rD1F28V4ITEM' 128,
device '/dev/rD1F29V1ITEM' 128,
device '/dev/rD1F29V2ITEM' 128,
device '/dev/rD1F29V3ITEM' 128,
device '/dev/rD1F29V4ITEM' 128,
device '/dev/rD1F30V1ITEM' 128,
device '/dev/rD1F30V2ITEM' 128,
device '/dev/rD1F30V3ITEM' 128,
device '/dev/rD1F30V4ITEM' 128,
device '/dev/rD1F31V1ITEM' 128,
device '/dev/rD1F31V2ITEM' 128,
device '/dev/rD1F31V3ITEM' 128,
device '/dev/rD1F31V4ITEM' 128,
device '/dev/rD1F32V1ITEM' 128,
device '/dev/rD1F32V2ITEM' 128,
device '/dev/rD1F32V3ITEM' 128,
device '/dev/rD1F32V4ITEM' 128,
device '/dev/rD1F33V1ITEM' 128,
device '/dev/rD1F33V2ITEM' 128,
device '/dev/rD1F33V3ITEM' 128,
device '/dev/rD1F33V4ITEM' 128,
device '/dev/rD1F34V1ITEM' 128,
device '/dev/rD1F34V2ITEM' 128,
device '/dev/rD1F34V3ITEM' 128,
device '/dev/rD1F34V4ITEM' 128,
device '/dev/rD1F35V1ITEM' 128,
device '/dev/rD1F35V2ITEM' 128,
device '/dev/rD1F35V3ITEM' 128,
device '/dev/rD1F35V4ITEM' 128,
device '/dev/rD1F36V1ITEM' 128,
device '/dev/rD1F36V2ITEM' 128,
device '/dev/rD1F36V3ITEM' 128,
device '/dev/rD1F36V4ITEM' 128,
device '/dev/rD1F37V1ITEM' 128,
device '/dev/rD1F37V2ITEM' 128,
device '/dev/rD1F37V3ITEM' 128,
device '/dev/rD1F37V4ITEM' 128,
device '/dev/rD1F38V1ITEM' 128,
device '/dev/rD1F38V2ITEM' 128,
device '/dev/rD1F38V3ITEM' 128,
device '/dev/rD1F38V4ITEM' 128,

```

```

device '/dev/rD1F39V1ITEM' 128,
device '/dev/rD1F39V2ITEM' 128,
device '/dev/rD1F39V3ITEM' 128,
device '/dev/rD1F39V4ITEM' 128,
device '/dev/rD1F40V1ITEM' 128,
device '/dev/rD1F40V2ITEM' 128,
device '/dev/rD1F40V3ITEM' 128,
device '/dev/rD1F40V4ITEM' 128
)
extentsize 16
prefetchsize 4096;
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' 148992,
device '/dev/rD1F01V2NORA' 148992,
device '/dev/rD1F01V3NORA' 148992,
device '/dev/rD1F01V4NORA' 148992,
device '/dev/rD1F02V1NORA' 148992,
device '/dev/rD1F02V2NORA' 148992,
device '/dev/rD1F02V3NORA' 148992,
device '/dev/rD1F02V4NORA' 148992,
device '/dev/rD1F03V1NORA' 148992,
device '/dev/rD1F03V2NORA' 148992,
device '/dev/rD1F03V3NORA' 148992,
device '/dev/rD1F03V4NORA' 148992,
device '/dev/rD1F04V1NORA' 148992,
device '/dev/rD1F04V2NORA' 148992,
device '/dev/rD1F04V3NORA' 148992,
device '/dev/rD1F04V4NORA' 148992,
device '/dev/rD1F05V1NORA' 148992,
device '/dev/rD1F05V2NORA' 148992,
device '/dev/rD1F05V3NORA' 148992,
device '/dev/rD1F05V4NORA' 148992
)
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/rD1F06V1NORA' 148992,
device '/dev/rD1F06V2NORA' 148992,
device '/dev/rD1F06V3NORA' 148992,
device '/dev/rD1F06V4NORA' 148992,
device '/dev/rD1F07V1NORA' 148992,
device '/dev/rD1F07V2NORA' 148992,
device '/dev/rD1F07V3NORA' 148992,
device '/dev/rD1F07V4NORA' 148992,

```

```

device '/dev/rD1F08V1NORA' 148992,
device '/dev/rD1F08V2NORA' 148992,
device '/dev/rD1F08V3NORA' 148992,
device '/dev/rD1F08V4NORA' 148992,
device '/dev/rD1F09V1NORA' 148992,
device '/dev/rD1F09V2NORA' 148992,
device '/dev/rD1F09V3NORA' 148992,
device '/dev/rD1F09V4NORA' 148992,
device '/dev/rD1F10V1NORA' 148992,
device '/dev/rD1F10V2NORA' 148992,
device '/dev/rD1F10V3NORA' 148992,
device '/dev/rD1F10V4NORA' 148992
)
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/rD1F11V1NORA' 148992,
device '/dev/rD1F11V2NORA' 148992,
device '/dev/rD1F11V3NORA' 148992,
device '/dev/rD1F11V4NORA' 148992,
device '/dev/rD1F12V1NORA' 148992,
device '/dev/rD1F12V2NORA' 148992,
device '/dev/rD1F12V3NORA' 148992,
device '/dev/rD1F12V4NORA' 148992,
device '/dev/rD1F13V1NORA' 148992,
device '/dev/rD1F13V2NORA' 148992,
device '/dev/rD1F13V3NORA' 148992,
device '/dev/rD1F13V4NORA' 148992,
device '/dev/rD1F14V1NORA' 148992,
device '/dev/rD1F14V2NORA' 148992,
device '/dev/rD1F14V3NORA' 148992,
device '/dev/rD1F14V4NORA' 148992,
device '/dev/rD1F15V1NORA' 148992,
device '/dev/rD1F15V2NORA' 148992,
device '/dev/rD1F15V3NORA' 148992,
device '/dev/rD1F15V4NORA' 148992
)
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/rD1F16V1NORA' 148992,
device '/dev/rD1F16V2NORA' 148992,
device '/dev/rD1F16V3NORA' 148992,
device '/dev/rD1F16V4NORA' 148992,
device '/dev/rD1F17V1NORA' 148992,
device '/dev/rD1F17V2NORA' 148992,
device '/dev/rD1F17V3NORA' 148992,
device '/dev/rD1F17V4NORA' 148992,
device '/dev/rD1F18V1NORA' 148992,
device '/dev/rD1F18V2NORA' 148992,

```

```

device '/dev/rD1F18V3NORA' 148992,
device '/dev/rD1F18V4NORA' 148992,
device '/dev/rD1F19V1NORA' 148992,
device '/dev/rD1F19V2NORA' 148992,
device '/dev/rD1F19V3NORA' 148992,
device '/dev/rD1F19V4NORA' 148992,
device '/dev/rD1F20V1NORA' 148992,
device '/dev/rD1F20V2NORA' 148992,
device '/dev/rD1F20V3NORA' 148992,
device '/dev/rD1F20V4NORA' 148992
)
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/rD1F21V1NORA' 148992,
device '/dev/rD1F21V2NORA' 148992,
device '/dev/rD1F21V3NORA' 148992,
device '/dev/rD1F21V4NORA' 148992,
device '/dev/rD1F22V1NORA' 148992,
device '/dev/rD1F22V2NORA' 148992,
device '/dev/rD1F22V3NORA' 148992,
device '/dev/rD1F22V4NORA' 148992,
device '/dev/rD1F23V1NORA' 148992,
device '/dev/rD1F23V2NORA' 148992,
device '/dev/rD1F23V3NORA' 148992,
device '/dev/rD1F23V4NORA' 148992,
device '/dev/rD1F24V1NORA' 148992,
device '/dev/rD1F24V2NORA' 148992,
device '/dev/rD1F24V3NORA' 148992,
device '/dev/rD1F24V4NORA' 148992,
device '/dev/rD1F25V1NORA' 148992,
device '/dev/rD1F25V2NORA' 148992,
device '/dev/rD1F25V3NORA' 148992,
device '/dev/rD1F25V4NORA' 148992
)
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/rD1F26V1NORA' 148992,
device '/dev/rD1F26V2NORA' 148992,
device '/dev/rD1F26V3NORA' 148992,
device '/dev/rD1F26V4NORA' 148992,
device '/dev/rD1F27V1NORA' 148992,
device '/dev/rD1F27V2NORA' 148992,
device '/dev/rD1F27V3NORA' 148992,
device '/dev/rD1F27V4NORA' 148992,
device '/dev/rD1F28V1NORA' 148992,
device '/dev/rD1F28V2NORA' 148992,
device '/dev/rD1F28V3NORA' 148992,
device '/dev/rD1F28V4NORA' 148992,

```

```

device '/dev/rD1F29V1NORA' 148992,
device '/dev/rD1F29V2NORA' 148992,
device '/dev/rD1F29V3NORA' 148992,
device '/dev/rD1F29V4NORA' 148992,
device '/dev/rD1F30V1NORA' 148992,
device '/dev/rD1F30V2NORA' 148992,
device '/dev/rD1F30V3NORA' 148992,
device '/dev/rD1F30V4NORA' 148992
)
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/rD1F31V1NORA' 148992,
device '/dev/rD1F31V2NORA' 148992,
device '/dev/rD1F31V3NORA' 148992,
device '/dev/rD1F31V4NORA' 148992,
device '/dev/rD1F32V1NORA' 148992,
device '/dev/rD1F32V2NORA' 148992,
device '/dev/rD1F32V3NORA' 148992,
device '/dev/rD1F32V4NORA' 148992,
device '/dev/rD1F33V1NORA' 148992,
device '/dev/rD1F33V2NORA' 148992,
device '/dev/rD1F33V3NORA' 148992,
device '/dev/rD1F33V4NORA' 148992,
device '/dev/rD1F34V1NORA' 148992,
device '/dev/rD1F34V2NORA' 148992,
device '/dev/rD1F34V3NORA' 148992,
device '/dev/rD1F34V4NORA' 148992,
device '/dev/rD1F35V1NORA' 148992,
device '/dev/rD1F35V2NORA' 148992,
device '/dev/rD1F35V3NORA' 148992,
device '/dev/rD1F35V4NORA' 148992
)
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/rD1F36V1NORA' 148992,
device '/dev/rD1F36V2NORA' 148992,
device '/dev/rD1F36V3NORA' 148992,
device '/dev/rD1F36V4NORA' 148992,
device '/dev/rD1F37V1NORA' 148992,
device '/dev/rD1F37V2NORA' 148992,
device '/dev/rD1F37V3NORA' 148992,
device '/dev/rD1F37V4NORA' 148992,
device '/dev/rD1F38V1NORA' 148992,
device '/dev/rD1F38V2NORA' 148992,
device '/dev/rD1F38V3NORA' 148992,
device '/dev/rD1F38V4NORA' 148992,
device '/dev/rD1F39V1NORA' 148992,
device '/dev/rD1F39V2NORA' 148992,

```

```

device '/dev/rD1F39V3NORA' 148992,
device '/dev/rD1F39V4NORA' 148992,
device '/dev/rD1F40V1NORA' 148992,
device '/dev/rD1F40V2NORA' 148992,
device '/dev/rD1F40V3NORA' 148992,
device '/dev/rD1F40V4NORA' 148992
)
extentsize 256
prefetchsize 4096;
commit;
connect reset;

ts/crts_newordb.ddl

connect to tpcc;
-- 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' 148992,
device '/dev/rD1F01V2NORB' 148992,
device '/dev/rD1F01V3NORB' 148992,
device '/dev/rD1F01V4NORB' 148992,
device '/dev/rD1F02V1NORB' 148992,
device '/dev/rD1F02V2NORB' 148992,
device '/dev/rD1F02V3NORB' 148992,
device '/dev/rD1F02V4NORB' 148992,
device '/dev/rD1F03V1NORB' 148992,
device '/dev/rD1F03V2NORB' 148992,
device '/dev/rD1F03V3NORB' 148992,
device '/dev/rD1F03V4NORB' 148992,
device '/dev/rD1F04V1NORB' 148992,
device '/dev/rD1F04V2NORB' 148992,
device '/dev/rD1F04V3NORB' 148992,
device '/dev/rD1F04V4NORB' 148992,
device '/dev/rD1F05V1NORB' 148992,
device '/dev/rD1F05V2NORB' 148992,
device '/dev/rD1F05V3NORB' 148992,
device '/dev/rD1F05V4NORB' 148992
)
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/rD1F06V1NORB' 148992,
device '/dev/rD1F06V2NORB' 148992,
device '/dev/rD1F06V3NORB' 148992,
device '/dev/rD1F06V4NORB' 148992,
device '/dev/rD1F07V1NORB' 148992,
device '/dev/rD1F07V2NORB' 148992,
device '/dev/rD1F07V3NORB' 148992,
device '/dev/rD1F07V4NORB' 148992,
device '/dev/rD1F08V1NORB' 148992,
device '/dev/rD1F08V2NORB' 148992,

```



```

        device '/dev/rD1F40V1NORB' 148992,
        device '/dev/rD1F40V2NORB' 148992,
        device '/dev/rD1F40V3NORB' 148992,
        device '/dev/rD1F40V4NORB' 148992
    )
    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' 291584
)
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' 291584
)
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' 291584
)
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' 291584
)
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/rD1F02V1ORD' 291584
)
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/rD1F02V2ORD' 291584
)
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/rD1F02V3ORD' 291584
)
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/rD1F02V4ORD' 291584
)
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/rD1F03V1ORD' 291584
)
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/rD1F03V2ORD' 291584
)
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/rD1F03V3ORD' 291584
)
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/rD1F03V4ORD' 291584
)
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/rD1F04V1ORD' 291584
)
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/rD1F04V2ORD' 291584
)
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/rD1F04V3ORD' 291584
)
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/rD1F04V4ORD' 291584
)
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/rD1F05V1ORD' 291584
)
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/rD1F05V2ORD' 291584
)
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/rD1F05V3ORD' 291584
)
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/rD1F05V4ORD' 291584
)
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/rD1F06V1ORD' 291584
)
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/rD1F06V2ORD' 291584
)
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/rD1F06V3ORD' 291584
)
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/rD1F06V4ORD' 291584
)
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/rD1F07V1ORD' 291584
)

```

```

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/rD1F07V2ORD' 291584
)
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/rD1F07V3ORD' 291584
)
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/rD1F07V4ORD' 291584
)
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/rD1F08V1ORD' 291584
)
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/rD1F08V2ORD' 291584
)
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/rD1F08V3ORD' 291584
)
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/rD1F08V4ORD' 291584
)
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/rD1F09V1ORD' 291584
)
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/rD1F09V2ORD' 291584
)
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/rD1F09V3ORD' 291584
)
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/rD1F09V4ORD' 291584
)
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/rD1F10V1ORD' 291584
)
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/rD1F10V2ORD' 291584
)
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/rD1F10V3ORD' 291584
)
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/rD1F10V4ORD' 291584
)
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/rD1F11V1ORD' 291584
)
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/rD1F11V2ORD' 291584
)
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/rD1F11V3ORD' 291584
)
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/rD1F11V4ORD' 291584
)
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/rD1F12V1ORD' 291584
)
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/rD1F12V2ORD' 291584
)
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/rD1F12V3ORD' 291584
)
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/rD1F12V4ORD' 291584
)
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/rD1F13V1ORD' 291584
)
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/rD1F13V2ORD' 291584
)
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/rD1F13V3ORD' 291584
)
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/rD1F13V4ORD' 291584
)
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/rD1F14V1ORD' 291584
)
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/rD1F14V2ORD' 291584
)
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/rD1F14V3ORD' 291584
)
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/rD1F14V4ORD' 291584
)

```

```

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/rD1F15V1ORD' 291584
)
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/rD1F15V2ORD' 291584
)
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/rD1F15V3ORD' 291584
)
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/rD1F15V4ORD' 291584
)
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/rD1F16V1ORD' 291584
)
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/rD1F16V2ORD' 291584
)
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/rD1F16V3ORD' 291584
)
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/rD1F16V4ORD' 291584
)
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/rD1F17V1ORD' 291584
)
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/rD1F17V2ORD' 291584
)
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/rD1F17V3ORD' 291584
)
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/rD1F17V4ORD' 291584
)
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/rD1F18V1ORD' 291584
)
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/rD1F18V2ORD' 291584
)
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/rD1F18V3ORD' 291584
)
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/rD1F18V4ORD' 291584
)
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/rD1F19V1ORD' 291584
)
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/rD1F19V2ORD' 291584
)
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/rD1F19V3ORD' 291584
)
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/rD1F19V4ORD' 291584
)
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/rD1F20V1ORD' 291584
)
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/rD1F20V2ORD' 291584
)
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/rD1F20V3ORD' 291584
)
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/rD1F20V4ORD' 291584
)
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/rD1F21V1ORD' 291584
)
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/rD1F21V2ORD' 291584
)
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/rD1F21V3ORD' 291584
)
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/rD1F21V4ORD' 291584
)
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/rD1F22V1ORD' 291584
)
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/rD1F22V2ORD' 291584
)
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/rD1F22V3ORD' 291584
)

```

```

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/rD1F22V4ORD' 291584
)
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/rD1F23V1ORD' 291584
)
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/rD1F23V2ORD' 291584
)
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/rD1F23V3ORD' 291584
)
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/rD1F23V4ORD' 291584
)
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/rD1F24V1ORD' 291584
)
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/rD1F24V2ORD' 291584
)
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/rD1F24V3ORD' 291584
)
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/rD1F24V4ORD' 291584
)
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/rD1F25V1ORD' 291584
)
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/rD1F25V2ORD' 291584
)
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/rD1F25V3ORD' 291584
)
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/rD1F25V4ORD' 291584
)
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/rD1F26V1ORD' 291584
)
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/rD1F26V2ORD' 291584
)
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/rD1F26V3ORD' 291584
)
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/rD1F26V4ORD' 291584
)
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/rD1F27V1ORD' 291584
)
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/rD1F27V2ORD' 291584
)
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/rD1F27V3ORD' 291584
)
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/rD1F27V4ORD' 291584
)
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/rD1F28V1ORD' 291584
)
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/rD1F28V2ORD' 291584
)
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/rD1F28V3ORD' 291584
)
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/rD1F28V4ORD' 291584
)
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/rD1F29V1ORD' 291584
)
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/rD1F29V2ORD' 291584
)
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/rD1F29V3ORD' 291584
)
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/rD1F29V4ORD' 291584
)
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/rD1F30V1ORD' 291584
)
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/rD1F30V2ORD' 291584
)

```

```

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/rD1F30V3ORD' 291584
)
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/rD1F30V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_121 of D1
drop tablespace ts_order_121;
create regular tablespace ts_order_121 pagesize 8K
managed by database
using
(
  device '/dev/rD1F31V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_122 of D1
drop tablespace ts_order_122;
create regular tablespace ts_order_122 pagesize 8K
managed by database
using
(
  device '/dev/rD1F31V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_123 of D1
drop tablespace ts_order_123;
create regular tablespace ts_order_123 pagesize 8K
managed by database
using
(
  device '/dev/rD1F31V3ORD' 291584
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_124 of D1
drop tablespace ts_order_124;
create regular tablespace ts_order_124 pagesize 8K
managed by database
using
(
    device '/dev/rD1F31V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_125 of D1
drop tablespace ts_order_125;
create regular tablespace ts_order_125 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_126 of D1
drop tablespace ts_order_126;
create regular tablespace ts_order_126 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_127 of D1
drop tablespace ts_order_127;
create regular tablespace ts_order_127 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_128 of D1
drop tablespace ts_order_128;
create regular tablespace ts_order_128 pagesize 8K
managed by database
using
(
    device '/dev/rD1F32V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_order_129 of D1
drop tablespace ts_order_129;
create regular tablespace ts_order_129 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_130 of D1
drop tablespace ts_order_130;
create regular tablespace ts_order_130 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_131 of D1
drop tablespace ts_order_131;
create regular tablespace ts_order_131 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_132 of D1
drop tablespace ts_order_132;
create regular tablespace ts_order_132 pagesize 8K
managed by database
using
(
    device '/dev/rD1F33V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_133 of D1
drop tablespace ts_order_133;
create regular tablespace ts_order_133 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_134 of D1
drop tablespace ts_order_134;

```

```

create regular tablespace ts_order_134 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_135 of D1
drop tablespace ts_order_135;
create regular tablespace ts_order_135 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_136 of D1
drop tablespace ts_order_136;
create regular tablespace ts_order_136 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_137 of D1
drop tablespace ts_order_137;
create regular tablespace ts_order_137 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_138 of D1
drop tablespace ts_order_138;
create regular tablespace ts_order_138 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_139 of D1
drop tablespace ts_order_139;
create regular tablespace ts_order_139 pagesize 8K
managed by database

```

```

using
(
  device '/dev/rD1F35V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_140 of D1
drop tablespace ts_order_140;
create regular tablespace ts_order_140 pagesize 8K
managed by database
using
(
  device '/dev/rD1F35V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_141 of D1
drop tablespace ts_order_141;
create regular tablespace ts_order_141 pagesize 8K
managed by database
using
(
  device '/dev/rD1F36V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_142 of D1
drop tablespace ts_order_142;
create regular tablespace ts_order_142 pagesize 8K
managed by database
using
(
  device '/dev/rD1F36V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_143 of D1
drop tablespace ts_order_143;
create regular tablespace ts_order_143 pagesize 8K
managed by database
using
(
  device '/dev/rD1F36V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_144 of D1
drop tablespace ts_order_144;
create regular tablespace ts_order_144 pagesize 8K
managed by database
using
(

```

```

  device '/dev/rD1F36V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_145 of D1
drop tablespace ts_order_145;
create regular tablespace ts_order_145 pagesize 8K
managed by database
using
(
  device '/dev/rD1F37V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_146 of D1
drop tablespace ts_order_146;
create regular tablespace ts_order_146 pagesize 8K
managed by database
using
(
  device '/dev/rD1F37V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_147 of D1
drop tablespace ts_order_147;
create regular tablespace ts_order_147 pagesize 8K
managed by database
using
(
  device '/dev/rD1F37V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_148 of D1
drop tablespace ts_order_148;
create regular tablespace ts_order_148 pagesize 8K
managed by database
using
(
  device '/dev/rD1F37V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_149 of D1
drop tablespace ts_order_149;
create regular tablespace ts_order_149 pagesize 8K
managed by database
using
(
  device '/dev/rD1F38V1ORD' 291584
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_150 of D1
drop tablespace ts_order_150;
create regular tablespace ts_order_150 pagesize 8K
managed by database
using
(
  device '/dev/rD1F38V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_151 of D1
drop tablespace ts_order_151;
create regular tablespace ts_order_151 pagesize 8K
managed by database
using
(
  device '/dev/rD1F38V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_152 of D1
drop tablespace ts_order_152;
create regular tablespace ts_order_152 pagesize 8K
managed by database
using
(
  device '/dev/rD1F38V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_153 of D1
drop tablespace ts_order_153;
create regular tablespace ts_order_153 pagesize 8K
managed by database
using
(
  device '/dev/rD1F39V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_154 of D1
drop tablespace ts_order_154;
create regular tablespace ts_order_154 pagesize 8K
managed by database
using
(
  device '/dev/rD1F39V2ORD' 291584
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_155 of D1
drop tablespace ts_order_155;
create regular tablespace ts_order_155 pagesize 8K
managed by database
using
(
device '/dev/rD1F39V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_156 of D1
drop tablespace ts_order_156;
create regular tablespace ts_order_156 pagesize 8K
managed by database
using
(
device '/dev/rD1F39V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_157 of D1
drop tablespace ts_order_157;
create regular tablespace ts_order_157 pagesize 8K
managed by database
using
(
device '/dev/rD1F40V1ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_158 of D1
drop tablespace ts_order_158;
create regular tablespace ts_order_158 pagesize 8K
managed by database
using
(
device '/dev/rD1F40V2ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_order_159 of D1
drop tablespace ts_order_159;
create regular tablespace ts_order_159 pagesize 8K
managed by database
using
(
device '/dev/rD1F40V3ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_order_160 of D1
drop tablespace ts_order_160;
create regular tablespace ts_order_160 pagesize 8K
managed by database
using
(
device '/dev/rD1F40V4ORD' 291584
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
connect reset;

```

ts/crts_orderline.ddl

```

connect to tpc;
-- 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' 7783936
)
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' 7783936
)
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' 7783936
)
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' 7783936
)

```

```

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/rD1F02V1ORL' 7783936
)
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/rD1F02V2ORL' 7783936
)
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/rD1F02V3ORL' 7783936
)
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/rD1F02V4ORL' 7783936
)
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/rD1F03V1ORL' 7783936
)
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/rD1F03V2ORL' 7783936
)
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/rD1F03V3ORL' 7783936
)
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/rD1F03V4ORL' 7783936
)
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/rD1F04V1ORL' 7783936
)
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/rD1F04V2ORL' 7783936
)
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/rD1F04V3ORL' 7783936
)
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/rD1F04V4ORL' 7783936
)
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/rD1F05V1ORL' 7783936
)
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/rD1F05V2ORL' 7783936
)
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/rD1F05V3ORL' 7783936
)
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/rD1F05V4ORL' 7783936
)
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/rD1F06V1ORL' 7783936
)
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/rD1F06V2ORL' 7783936
)
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/rD1F06V3ORL' 7783936
)
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/rD1F06V4ORL' 7783936
)
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/rD1F07V1ORL' 7783936
)
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/rD1F07V2ORL' 7783936
)
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/rD1F07V3ORL' 7783936
)
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/rD1F07V4ORL' 7783936
)
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/rD1F08V1ORL' 7783936
)
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/rD1F08V2ORL' 7783936
)
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/rD1F08V3ORL' 7783936
)
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/rD1F08V4ORL' 7783936
)
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/rD1F09V1ORL' 7783936
)
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/rD1F09V2ORL' 7783936
)
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/rD1F09V3ORL' 7783936
)

```

```

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/rD1F09V4ORL' 7783936
)
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/rD1F10V1ORL' 7783936
)
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/rD1F10V2ORL' 7783936
)
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/rD1F10V3ORL' 7783936
)
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/rD1F10V4ORL' 7783936
)
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/rD1F11V1ORL' 7783936
)
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/rD1F11V2ORL' 7783936
)
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/rD1F11V3ORL' 7783936
)
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/rD1F11V4ORL' 7783936
)
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/rD1F12V1ORL' 7783936
)
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/rD1F12V2ORL' 7783936
)
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/rD1F12V3ORL' 7783936
)
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/rD1F12V4ORL' 7783936
)
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/rD1F13V1ORL' 7783936
)
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/rD1F13V2ORL' 7783936
)
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/rD1F13V3ORL' 7783936
)
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/rD1F13V4ORL' 7783936
)
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/rD1F14V1ORL' 7783936
)
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/rD1F14V2ORL' 7783936
)
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/rD1F14V3ORL' 7783936
)
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/rD1F14V4ORL' 7783936
)
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/rD1F15V1ORL' 7783936
)
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/rD1F15V2ORL' 7783936
)
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/rD1F15V3ORL' 7783936
)
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/rD1F15V4ORL' 7783936
)
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/rD1F16V1ORL' 7783936
)
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/rD1F16V2ORL' 7783936
)
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/rD1F16V3ORL' 7783936
)
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/rD1F16V4ORL' 7783936
)
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/rD1F17V1ORL' 7783936
)
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/rD1F17V2ORL' 7783936
)

```

```

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/rD1F17V3ORL' 7783936
)
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/rD1F17V4ORL' 7783936
)
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/rD1F18V1ORL' 7783936
)
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/rD1F18V2ORL' 7783936
)
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/rD1F18V3ORL' 7783936
)
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/rD1F18V4ORL' 7783936
)
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/rD1F19V1ORL' 7783936
)
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/rD1F19V2ORL' 7783936
)
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/rD1F19V3ORL' 7783936
)
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/rD1F19V4ORL' 7783936
)
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/rD1F20V1ORL' 7783936
)
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/rD1F20V2ORL' 7783936
)
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/rD1F20V3ORL' 7783936
)
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/rD1F20V4ORL' 7783936
)
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/rD1F21V1ORL' 7783936
)
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/rD1F21V2ORL' 7783936
)
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/rD1F21V3ORL' 7783936
)
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/rD1F21V4ORL' 7783936
)
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/rD1F22V1ORL' 7783936
)
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/rD1F22V2ORL' 7783936
)
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/rD1F22V3ORL' 7783936
)
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/rD1F22V4ORL' 7783936
)
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/rD1F23V1ORL' 7783936
)
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/rD1F23V2ORL' 7783936
)
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/rD1F23V3ORL' 7783936
)
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/rD1F23V4ORL' 7783936
)
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/rD1F24V1ORL' 7783936
)
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/rD1F24V2ORL' 7783936
)
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/rD1F24V3ORL' 7783936
)
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/rD1F24V4ORL' 7783936
)
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/rD1F25V1ORL' 7783936
)

```

```

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/rD1F25V2ORL' 7783936
)
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/rD1F25V3ORL' 7783936
)
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/rD1F25V4ORL' 7783936
)
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/rD1F26V1ORL' 7783936
)
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/rD1F26V2ORL' 7783936
)
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/rD1F26V3ORL' 7783936
)
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/rD1F26V4ORL' 7783936
)
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/rD1F27V1ORL' 7783936
)
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/rD1F27V2ORL' 7783936
)
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/rD1F27V3ORL' 7783936
)
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/rD1F27V4ORL' 7783936
)
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/rD1F28V1ORL' 7783936
)
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/rD1F28V2ORL' 7783936
)
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/rD1F28V3ORL' 7783936
)
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/rD1F28V4ORL' 7783936
)
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/rD1F29V1ORL' 7783936
)
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/rD1F29V2ORL' 7783936
)
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/rD1F29V3ORL' 7783936
)
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/rD1F29V4ORL' 7783936
)
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/rD1F30V1ORL' 7783936
)
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/rD1F30V2ORL' 7783936
)
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/rD1F30V3ORL' 7783936
)
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/rD1F30V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_121 of D1
drop tablespace ts_orderline_121;
create regular tablespace ts_orderline_121 pagesize 8K
managed by database
using
(
  device '/dev/rD1F31V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_122 of D1
drop tablespace ts_orderline_122;
create regular tablespace ts_orderline_122 pagesize 8K
managed by database
using
(
  device '/dev/rD1F31V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_123 of D1
drop tablespace ts_orderline_123;
create regular tablespace ts_orderline_123 pagesize 8K
managed by database
using
(

```

```

  device '/dev/rD1F31V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_124 of D1
drop tablespace ts_orderline_124;
create regular tablespace ts_orderline_124 pagesize 8K
managed by database
using
(
  device '/dev/rD1F31V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_125 of D1
drop tablespace ts_orderline_125;
create regular tablespace ts_orderline_125 pagesize 8K
managed by database
using
(
  device '/dev/rD1F32V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_126 of D1
drop tablespace ts_orderline_126;
create regular tablespace ts_orderline_126 pagesize 8K
managed by database
using
(
  device '/dev/rD1F32V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_127 of D1
drop tablespace ts_orderline_127;
create regular tablespace ts_orderline_127 pagesize 8K
managed by database
using
(
  device '/dev/rD1F32V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_128 of D1
drop tablespace ts_orderline_128;
create regular tablespace ts_orderline_128 pagesize 8K
managed by database
using
(
  device '/dev/rD1F32V4ORL' 7783936
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_129 of D1
drop tablespace ts_orderline_129;
create regular tablespace ts_orderline_129 pagesize 8K
managed by database
using
(
  device '/dev/rD1F33V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_130 of D1
drop tablespace ts_orderline_130;
create regular tablespace ts_orderline_130 pagesize 8K
managed by database
using
(
  device '/dev/rD1F33V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_131 of D1
drop tablespace ts_orderline_131;
create regular tablespace ts_orderline_131 pagesize 8K
managed by database
using
(
  device '/dev/rD1F33V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_132 of D1
drop tablespace ts_orderline_132;
create regular tablespace ts_orderline_132 pagesize 8K
managed by database
using
(
  device '/dev/rD1F33V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_133 of D1
drop tablespace ts_orderline_133;
create regular tablespace ts_orderline_133 pagesize 8K
managed by database
using
(
  device '/dev/rD1F34V1ORL' 7783936
)
extentsize 256
prefetchsize 4096

```



```

bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_134 of D1
drop tablespace ts_orderline_134;
create regular tablespace ts_orderline_134 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_135 of D1
drop tablespace ts_orderline_135;
create regular tablespace ts_orderline_135 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_136 of D1
drop tablespace ts_orderline_136;
create regular tablespace ts_orderline_136 pagesize 8K
managed by database
using
(
    device '/dev/rD1F34V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_137 of D1
drop tablespace ts_orderline_137;
create regular tablespace ts_orderline_137 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_138 of D1
drop tablespace ts_orderline_138;
create regular tablespace ts_orderline_138 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_orderline_139 of D1
drop tablespace ts_orderline_139;
create regular tablespace ts_orderline_139 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_140 of D1
drop tablespace ts_orderline_140;
create regular tablespace ts_orderline_140 pagesize 8K
managed by database
using
(
    device '/dev/rD1F35V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_141 of D1
drop tablespace ts_orderline_141;
create regular tablespace ts_orderline_141 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_142 of D1
drop tablespace ts_orderline_142;
create regular tablespace ts_orderline_142 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_143 of D1
drop tablespace ts_orderline_143;
create regular tablespace ts_orderline_143 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_144 of D1
drop tablespace ts_orderline_144;

```

```

create regular tablespace ts_orderline_144 pagesize 8K
managed by database
using
(
    device '/dev/rD1F36V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_145 of D1
drop tablespace ts_orderline_145;
create regular tablespace ts_orderline_145 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_146 of D1
drop tablespace ts_orderline_146;
create regular tablespace ts_orderline_146 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_147 of D1
drop tablespace ts_orderline_147;
create regular tablespace ts_orderline_147 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_148 of D1
drop tablespace ts_orderline_148;
create regular tablespace ts_orderline_148 pagesize 8K
managed by database
using
(
    device '/dev/rD1F37V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_149 of D1
drop tablespace ts_orderline_149;
create regular tablespace ts_orderline_149 pagesize 8K
managed by database

```

```

using
(
  device '/dev/rD1F38V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_150 of D1
drop tablespace ts_orderline_150;
create regular tablespace ts_orderline_150 pagesize 8K
managed by database
using
(
  device '/dev/rD1F38V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_151 of D1
drop tablespace ts_orderline_151;
create regular tablespace ts_orderline_151 pagesize 8K
managed by database
using
(
  device '/dev/rD1F38V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_152 of D1
drop tablespace ts_orderline_152;
create regular tablespace ts_orderline_152 pagesize 8K
managed by database
using
(
  device '/dev/rD1F38V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_153 of D1
drop tablespace ts_orderline_153;
create regular tablespace ts_orderline_153 pagesize 8K
managed by database
using
(
  device '/dev/rD1F39V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_154 of D1
drop tablespace ts_orderline_154;
create regular tablespace ts_orderline_154 pagesize 8K
managed by database
using
(

```

```

device '/dev/rD1F39V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_155 of D1
drop tablespace ts_orderline_155;
create regular tablespace ts_orderline_155 pagesize 8K
managed by database
using
(
  device '/dev/rD1F39V3ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_156 of D1
drop tablespace ts_orderline_156;
create regular tablespace ts_orderline_156 pagesize 8K
managed by database
using
(
  device '/dev/rD1F39V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_157 of D1
drop tablespace ts_orderline_157;
create regular tablespace ts_orderline_157 pagesize 8K
managed by database
using
(
  device '/dev/rD1F40V1ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_158 of D1
drop tablespace ts_orderline_158;
create regular tablespace ts_orderline_158 pagesize 8K
managed by database
using
(
  device '/dev/rD1F40V2ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_159 of D1
drop tablespace ts_orderline_159;
create regular tablespace ts_orderline_159 pagesize 8K
managed by database
using
(
  device '/dev/rD1F40V3ORL' 7783936
)

```

```

extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
-- now creating TS for ts_orderline_160 of D1
drop tablespace ts_orderline_160;
create regular tablespace ts_orderline_160 pagesize 8K
managed by database
using
(
  device '/dev/rD1F40V4ORL' 7783936
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;
connect reset;

```

ts/crts stock.ddl

```

connect to tpcc;
-- 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' 14001664
)
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' 14001664
)
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' 14001664
)
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' 14001664
)

```

```

    )
    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/rD1F02V1STK' 14001664
)
    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/rD1F02V2STK' 14001664
)
    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/rD1F02V3STK' 14001664
)
    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/rD1F02V4STK' 14001664
)
    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/rD1F03V1STK' 14001664
)
    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/rD1F03V2STK' 14001664
)
    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/rD1F03V3STK' 14001664
)
    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/rD1F03V4STK' 14001664
)
    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/rD1F04V1STK' 14001664
)
    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/rD1F04V2STK' 14001664
)
    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/rD1F04V3STK' 14001664
)
    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/rD1F04V4STK' 14001664
)
    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/rD1F05V1STK' 14001664
)
    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/rD1F05V2STK' 14001664
)
    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/rD1F05V3STK' 14001664
)
    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/rD1F05V4STK' 14001664
)
    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/rD1F06V1STK' 14001664
    )
    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/rD1F06V2STK' 14001664
)
    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/rD1F06V3STK' 14001664
)
    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/rD1F06V4STK' 14001664
)
    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/rD1F07V1STK' 14001664
)
    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/rD1F07V2STK' 14001664
)
    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/rD1F07V3STK' 14001664
)
    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/rD1F07V4STK' 14001664
)
    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/rD1F08V1STK' 14001664
)
    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/rD1F08V2STK' 14001664
)
    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/rD1F08V3STK' 14001664
)
    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/rD1F08V4STK' 14001664
)
    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/rD1F09V1STK' 14001664
)
    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/rD1F09V2STK' 14001664
)
    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/rD1F09V3STK' 14001664
)
    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/rD1F09V4STK' 14001664
)
    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/rD1F10V1STK' 14001664
)
    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/rD1F10V2STK' 14001664
)
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/rD1F10V3STK' 14001664
)
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/rD1F10V4STK' 14001664
)
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/rD1F11V1STK' 14001664
)
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/rD1F11V2STK' 14001664
)
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/rD1F11V3STK' 14001664
)
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/rD1F11V4STK' 14001664
)
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/rD1F12V1STK' 14001664
)
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/rD1F12V2STK' 14001664
)
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/rD1F12V3STK' 14001664
)
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/rD1F12V4STK' 14001664
)
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/rD1F13V1STK' 14001664
)

```

```

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/rD1F13V2STK' 14001664
)
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/rD1F13V3STK' 14001664
)
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/rD1F13V4STK' 14001664
)
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/rD1F14V1STK' 14001664
)
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/rD1F14V2STK' 14001664
)
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/rD1F14V3STK' 14001664
)
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/rD1F14V4STK' 14001664
)
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/rD1F15V1STK' 14001664
)
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/rD1F15V2STK' 14001664
)
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/rD1F15V3STK' 14001664
)
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/rD1F15V4STK' 14001664
)
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/rD1F16V1STK' 14001664
)
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/rD1F16V2STK' 14001664
)
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/rD1F16V3STK' 14001664
)
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/rD1F16V4STK' 14001664
)
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/rD1F17V1STK' 14001664
)
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/rD1F17V2STK' 14001664
)

```

```

)
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/rD1F17V3STK' 14001664
)
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/rD1F17V4STK' 14001664
)
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/rD1F18V1STK' 14001664
)
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/rD1F18V2STK' 14001664
)
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/rD1F18V3STK' 14001664
)
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/rD1F18V4STK' 14001664
)
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/rD1F19V1STK' 14001664
)
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/rD1F19V2STK' 14001664
)
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/rD1F19V3STK' 14001664
)
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/rD1F19V4STK' 14001664
)
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/rD1F20V1STK' 14001664
)
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/rD1F20V2STK' 14001664
)
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/rD1F20V3STK' 14001664
)
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/rD1F20V4STK' 14001664
)
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/rD1F21V1STK' 14001664
)
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/rD1F21V2STK' 14001664
)
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/rD1F21V3STK' 14001664
)
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/rD1F21V4STK' 14001664
)
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/rD1F22V1STK' 14001664
)
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/rD1F22V2STK' 14001664
)
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/rD1F22V3STK' 14001664
)
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/rD1F22V4STK' 14001664
)
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/rD1F23V1STK' 14001664
)
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/rD1F23V2STK' 14001664
)
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/rD1F23V3STK' 14001664
)
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/rD1F23V4STK' 14001664
)
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/rD1F24V1STK' 14001664
)
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/rD1F24V2STK' 14001664
)
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/rD1F24V3STK' 14001664
)
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/rD1F24V4STK' 14001664
)
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/rD1F25V1STK' 14001664
)
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/rD1F25V2STK' 14001664
)
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/rD1F25V3STK' 14001664
)
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/rD1F25V4STK' 14001664
)
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/rD1F26V1STK' 14001664
)
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/rD1F26V2STK' 14001664
)
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/rD1F26V3STK' 14001664
)
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/rD1F26V4STK' 14001664
)
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/rD1F27V1STK' 14001664
)
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/rD1F27V2STK' 14001664
)
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/rD1F27V3STK' 14001664
)
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/rD1F27V4STK' 14001664
)
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/rD1F28V1STK' 14001664
)
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/rD1F28V2STK' 14001664
)
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/rD1F28V3STK' 14001664
)

```

```

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/rD1F28V4STK' 14001664
)
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/rD1F29V1STK' 14001664
)
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/rD1F29V2STK' 14001664
)
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/rD1F29V3STK' 14001664
)
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/rD1F29V4STK' 14001664
)
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/rD1F30V1STK' 14001664
)
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/rD1F30V2STK' 14001664
)
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/rD1F30V3STK' 14001664
)
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/rD1F30V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_121 of D1
drop tablespace ts_stock_121;
create regular tablespace ts_stock_121 pagesize 4K
managed by database
using
(
    device '/dev/rD1F31V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_122 of D1
drop tablespace ts_stock_122;
create regular tablespace ts_stock_122 pagesize 4K
managed by database
using
(
    device '/dev/rD1F31V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;

```

```

-- now creating TS for ts_stock_123 of D1
drop tablespace ts_stock_123;
create regular tablespace ts_stock_123 pagesize 4K
managed by database
using
(
    device '/dev/rD1F31V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_124 of D1
drop tablespace ts_stock_124;
create regular tablespace ts_stock_124 pagesize 4K
managed by database
using
(
    device '/dev/rD1F31V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_125 of D1
drop tablespace ts_stock_125;
create regular tablespace ts_stock_125 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_126 of D1
drop tablespace ts_stock_126;
create regular tablespace ts_stock_126 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_127 of D1
drop tablespace ts_stock_127;
create regular tablespace ts_stock_127 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_128 of D1
drop tablespace ts_stock_128;
create regular tablespace ts_stock_128 pagesize 4K
managed by database
using
(
    device '/dev/rD1F32V4STK' 14001664

```

```

)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_129 of D1
drop tablespace ts_stock_129;
create regular tablespace ts_stock_129 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_130 of D1
drop tablespace ts_stock_130;
create regular tablespace ts_stock_130 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_131 of D1
drop tablespace ts_stock_131;
create regular tablespace ts_stock_131 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_132 of D1
drop tablespace ts_stock_132;
create regular tablespace ts_stock_132 pagesize 4K
managed by database
using
(
    device '/dev/rD1F33V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_133 of D1
drop tablespace ts_stock_133;
create regular tablespace ts_stock_133 pagesize 4K
managed by database
using
(
    device '/dev/rD1F34V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_134 of D1
drop tablespace ts_stock_134;
create regular tablespace ts_stock_134 pagesize 4K

```

```

managed by database
using
(
    device '/dev/rD1F34V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_135 of D1
drop tablespace ts_stock_135;
create regular tablespace ts_stock_135 pagesize 4K
managed by database
using
(
    device '/dev/rD1F34V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_136 of D1
drop tablespace ts_stock_136;
create regular tablespace ts_stock_136 pagesize 4K
managed by database
using
(
    device '/dev/rD1F34V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_137 of D1
drop tablespace ts_stock_137;
create regular tablespace ts_stock_137 pagesize 4K
managed by database
using
(
    device '/dev/rD1F35V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_138 of D1
drop tablespace ts_stock_138;
create regular tablespace ts_stock_138 pagesize 4K
managed by database
using
(
    device '/dev/rD1F35V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_139 of D1
drop tablespace ts_stock_139;
create regular tablespace ts_stock_139 pagesize 4K
managed by database
using
(
    device '/dev/rD1F35V3STK' 14001664
)
extentsize 256
prefetchsize 4096;

```

```

commit;
-- now creating TS for ts_stock_140 of D1
drop tablespace ts_stock_140;
create regular tablespace ts_stock_140 pagesize 4K
managed by database
using
(
    device '/dev/rD1F35V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_141 of D1
drop tablespace ts_stock_141;
create regular tablespace ts_stock_141 pagesize 4K
managed by database
using
(
    device '/dev/rD1F36V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_142 of D1
drop tablespace ts_stock_142;
create regular tablespace ts_stock_142 pagesize 4K
managed by database
using
(
    device '/dev/rD1F36V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_143 of D1
drop tablespace ts_stock_143;
create regular tablespace ts_stock_143 pagesize 4K
managed by database
using
(
    device '/dev/rD1F36V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_144 of D1
drop tablespace ts_stock_144;
create regular tablespace ts_stock_144 pagesize 4K
managed by database
using
(
    device '/dev/rD1F36V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_145 of D1
drop tablespace ts_stock_145;
create regular tablespace ts_stock_145 pagesize 4K
managed by database
using
(

```

```

    device '/dev/rD1F37V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_146 of D1
drop tablespace ts_stock_146;
create regular tablespace ts_stock_146 pagesize 4K
managed by database
using
(
    device '/dev/rD1F37V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_147 of D1
drop tablespace ts_stock_147;
create regular tablespace ts_stock_147 pagesize 4K
managed by database
using
(
    device '/dev/rD1F37V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_148 of D1
drop tablespace ts_stock_148;
create regular tablespace ts_stock_148 pagesize 4K
managed by database
using
(
    device '/dev/rD1F37V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_149 of D1
drop tablespace ts_stock_149;
create regular tablespace ts_stock_149 pagesize 4K
managed by database
using
(
    device '/dev/rD1F38V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_150 of D1
drop tablespace ts_stock_150;
create regular tablespace ts_stock_150 pagesize 4K
managed by database
using
(
    device '/dev/rD1F38V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_151 of D1
drop tablespace ts_stock_151;

```

```

create regular tablespace ts_stock_151 pagesize 4K
managed by database
using
(
    device '/dev/rD1F38V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_152 of D1
drop tablespace ts_stock_152;
create regular tablespace ts_stock_152 pagesize 4K
managed by database
using
(
    device '/dev/rD1F38V4STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_153 of D1
drop tablespace ts_stock_153;
create regular tablespace ts_stock_153 pagesize 4K
managed by database
using
(
    device '/dev/rD1F39V1STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_154 of D1
drop tablespace ts_stock_154;
create regular tablespace ts_stock_154 pagesize 4K
managed by database
using
(
    device '/dev/rD1F39V2STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_155 of D1
drop tablespace ts_stock_155;
create regular tablespace ts_stock_155 pagesize 4K
managed by database
using
(
    device '/dev/rD1F39V3STK' 14001664
)
extentsize 256
prefetchsize 4096;
commit;
-- now creating TS for ts_stock_156 of D1
drop tablespace ts_stock_156;
create regular tablespace ts_stock_156 pagesize 4K
managed by database
using
(
    device '/dev/rD1F39V4STK' 14001664
)
extentsize 256

```

```

    prefetchsize 4096;
commit;
-- now creating TS for ts_stock_157 of D1
drop tablespace ts_stock_157;
create regular tablespace ts_stock_157 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V1STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;
-- now creating TS for ts_stock_158 of D1
drop tablespace ts_stock_158;
create regular tablespace ts_stock_158 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V2STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;
-- now creating TS for ts_stock_159 of D1
drop tablespace ts_stock_159;
create regular tablespace ts_stock_159 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V3STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;
-- now creating TS for ts_stock_160 of D1
drop tablespace ts_stock_160;
create regular tablespace ts_stock_160 pagesize 4K
managed by database
using
(
    device '/dev/rD1F40V4STK' 14001664
)
    extentsize 256
    prefetchsize 4096;
commit;
connect reset;

```

ts/crts_ware.ddl

```

connect to tpcc;
-- 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' 2048,
    device '/dev/rD1F01V2WARE' 2048,
    device '/dev/rD1F01V3WARE' 2048,
    device '/dev/rD1F01V4WARE' 2048,

```

```

    device '/dev/rD1F02V1WARE' 2048,
    device '/dev/rD1F02V2WARE' 2048,
    device '/dev/rD1F02V3WARE' 2048,
    device '/dev/rD1F02V4WARE' 2048,
    device '/dev/rD1F03V1WARE' 2048,
    device '/dev/rD1F03V2WARE' 2048,
    device '/dev/rD1F03V3WARE' 2048,
    device '/dev/rD1F03V4WARE' 2048,
    device '/dev/rD1F04V1WARE' 2048,
    device '/dev/rD1F04V2WARE' 2048,
    device '/dev/rD1F04V3WARE' 2048,
    device '/dev/rD1F04V4WARE' 2048,
    device '/dev/rD1F05V1WARE' 2048,
    device '/dev/rD1F05V2WARE' 2048,
    device '/dev/rD1F05V3WARE' 2048,
    device '/dev/rD1F05V4WARE' 2048
)
    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/rD1F06V1WARE' 2048,
    device '/dev/rD1F06V2WARE' 2048,
    device '/dev/rD1F06V3WARE' 2048,
    device '/dev/rD1F06V4WARE' 2048,
    device '/dev/rD1F07V1WARE' 2048,
    device '/dev/rD1F07V2WARE' 2048,
    device '/dev/rD1F07V3WARE' 2048,
    device '/dev/rD1F07V4WARE' 2048,
    device '/dev/rD1F08V1WARE' 2048,
    device '/dev/rD1F08V2WARE' 2048,
    device '/dev/rD1F08V3WARE' 2048,
    device '/dev/rD1F08V4WARE' 2048,
    device '/dev/rD1F09V1WARE' 2048,
    device '/dev/rD1F09V2WARE' 2048,
    device '/dev/rD1F09V3WARE' 2048,
    device '/dev/rD1F09V4WARE' 2048,
    device '/dev/rD1F10V1WARE' 2048,
    device '/dev/rD1F10V2WARE' 2048,
    device '/dev/rD1F10V3WARE' 2048,
    device '/dev/rD1F10V4WARE' 2048
)
    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/rD1F11V1WARE' 2048,
    device '/dev/rD1F11V2WARE' 2048,
    device '/dev/rD1F11V3WARE' 2048,
    device '/dev/rD1F11V4WARE' 2048,
    device '/dev/rD1F12V1WARE' 2048,
    device '/dev/rD1F12V2WARE' 2048,

```

```

    device '/dev/rD1F12V3WARE' 2048,
    device '/dev/rD1F12V4WARE' 2048,
    device '/dev/rD1F13V1WARE' 2048,
    device '/dev/rD1F13V2WARE' 2048,
    device '/dev/rD1F13V3WARE' 2048,
    device '/dev/rD1F13V4WARE' 2048,
    device '/dev/rD1F14V1WARE' 2048,
    device '/dev/rD1F14V2WARE' 2048,
    device '/dev/rD1F14V3WARE' 2048,
    device '/dev/rD1F14V4WARE' 2048,
    device '/dev/rD1F15V1WARE' 2048,
    device '/dev/rD1F15V2WARE' 2048,
    device '/dev/rD1F15V3WARE' 2048,
    device '/dev/rD1F15V4WARE' 2048
)
    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/rD1F16V1WARE' 2048,
    device '/dev/rD1F16V2WARE' 2048,
    device '/dev/rD1F16V3WARE' 2048,
    device '/dev/rD1F16V4WARE' 2048,
    device '/dev/rD1F17V1WARE' 2048,
    device '/dev/rD1F17V2WARE' 2048,
    device '/dev/rD1F17V3WARE' 2048,
    device '/dev/rD1F17V4WARE' 2048,
    device '/dev/rD1F18V1WARE' 2048,
    device '/dev/rD1F18V2WARE' 2048,
    device '/dev/rD1F18V3WARE' 2048,
    device '/dev/rD1F18V4WARE' 2048,
    device '/dev/rD1F19V1WARE' 2048,
    device '/dev/rD1F19V2WARE' 2048,
    device '/dev/rD1F19V3WARE' 2048,
    device '/dev/rD1F19V4WARE' 2048,
    device '/dev/rD1F20V1WARE' 2048,
    device '/dev/rD1F20V2WARE' 2048,
    device '/dev/rD1F20V3WARE' 2048,
    device '/dev/rD1F20V4WARE' 2048
)
    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/rD1F21V1WARE' 2048,
    device '/dev/rD1F21V2WARE' 2048,
    device '/dev/rD1F21V3WARE' 2048,
    device '/dev/rD1F21V4WARE' 2048,
    device '/dev/rD1F22V1WARE' 2048,
    device '/dev/rD1F22V2WARE' 2048,
    device '/dev/rD1F22V3WARE' 2048,
    device '/dev/rD1F22V4WARE' 2048,

```

```

device '/dev/rD1F23V1WARE' 2048,
device '/dev/rD1F23V2WARE' 2048,
device '/dev/rD1F23V3WARE' 2048,
device '/dev/rD1F23V4WARE' 2048,
device '/dev/rD1F24V1WARE' 2048,
device '/dev/rD1F24V2WARE' 2048,
device '/dev/rD1F24V3WARE' 2048,
device '/dev/rD1F24V4WARE' 2048,
device '/dev/rD1F25V1WARE' 2048,
device '/dev/rD1F25V2WARE' 2048,
device '/dev/rD1F25V3WARE' 2048,
device '/dev/rD1F25V4WARE' 2048
)
)
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/rD1F26V1WARE' 2048,
device '/dev/rD1F26V2WARE' 2048,
device '/dev/rD1F26V3WARE' 2048,
device '/dev/rD1F26V4WARE' 2048,
device '/dev/rD1F27V1WARE' 2048,
device '/dev/rD1F27V2WARE' 2048,
device '/dev/rD1F27V3WARE' 2048,
device '/dev/rD1F27V4WARE' 2048,
device '/dev/rD1F28V1WARE' 2048,
device '/dev/rD1F28V2WARE' 2048,
device '/dev/rD1F28V3WARE' 2048,
device '/dev/rD1F28V4WARE' 2048,
device '/dev/rD1F29V1WARE' 2048,
device '/dev/rD1F29V2WARE' 2048,
device '/dev/rD1F29V3WARE' 2048,
device '/dev/rD1F29V4WARE' 2048,
device '/dev/rD1F30V1WARE' 2048,
device '/dev/rD1F30V2WARE' 2048,
device '/dev/rD1F30V3WARE' 2048,
device '/dev/rD1F30V4WARE' 2048
)
)
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/rD1F31V1WARE' 2048,
device '/dev/rD1F31V2WARE' 2048,
device '/dev/rD1F31V3WARE' 2048,
device '/dev/rD1F31V4WARE' 2048,
device '/dev/rD1F32V1WARE' 2048,
device '/dev/rD1F32V2WARE' 2048,
device '/dev/rD1F32V3WARE' 2048,
device '/dev/rD1F32V4WARE' 2048,
device '/dev/rD1F33V1WARE' 2048,
device '/dev/rD1F33V2WARE' 2048,

```

```

device '/dev/rD1F33V3WARE' 2048,
device '/dev/rD1F33V4WARE' 2048,
device '/dev/rD1F34V1WARE' 2048,
device '/dev/rD1F34V2WARE' 2048,
device '/dev/rD1F34V3WARE' 2048,
device '/dev/rD1F34V4WARE' 2048,
device '/dev/rD1F35V1WARE' 2048,
device '/dev/rD1F35V2WARE' 2048,
device '/dev/rD1F35V3WARE' 2048,
device '/dev/rD1F35V4WARE' 2048
)
)
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/rD1F36V1WARE' 2048,
device '/dev/rD1F36V2WARE' 2048,
device '/dev/rD1F36V3WARE' 2048,
device '/dev/rD1F36V4WARE' 2048,
device '/dev/rD1F37V1WARE' 2048,
device '/dev/rD1F37V2WARE' 2048,
device '/dev/rD1F37V3WARE' 2048,
device '/dev/rD1F37V4WARE' 2048,
device '/dev/rD1F38V1WARE' 2048,
device '/dev/rD1F38V2WARE' 2048,
device '/dev/rD1F38V3WARE' 2048,
device '/dev/rD1F38V4WARE' 2048,
device '/dev/rD1F39V1WARE' 2048,
device '/dev/rD1F39V2WARE' 2048,
device '/dev/rD1F39V3WARE' 2048,
device '/dev/rD1F39V4WARE' 2048,
device '/dev/rD1F40V1WARE' 2048,
device '/dev/rD1F40V2WARE' 2048,
device '/dev/rD1F40V3WARE' 2048,
device '/dev/rD1F40V4WARE' 2048
)
)
extentsize 32
prefetchsize 4096;
commit;
connect reset;

```

bp/alter 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
-- 2004
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.

```

```

-----
-- Alter Size of Bufferpools
connect to tpcc;
alter bufferpool IBMDEFAULTBP DEFERRED size 1000;
alter bufferpool IBMDEFAULTBP8K DEFERRED size 16;
alter bufferpool IBMDEFAULTBP16K DEFERRED size 16;
alter bufferpool WAR1 DEFERRED size 1800;
alter bufferpool WAR2 DEFERRED size 1800;
alter bufferpool WAR3 DEFERRED size 1800;
alter bufferpool WAR4 DEFERRED size 1800;
alter bufferpool WAR5 DEFERRED size 1800;
alter bufferpool WAR6 DEFERRED size 1800;
alter bufferpool WAR7 DEFERRED size 1800;
alter bufferpool WAR8 DEFERRED size 1800;
alter bufferpool DIS1 DEFERRED size 9250;
alter bufferpool DIS2 DEFERRED size 9250;
alter bufferpool DIS3 DEFERRED size 9250;
alter bufferpool DIS4 DEFERRED size 9250;
alter bufferpool DIS5 DEFERRED size 9250;
alter bufferpool DIS6 DEFERRED size 9250;
alter bufferpool DIS7 DEFERRED size 9250;
alter bufferpool DIS8 DEFERRED size 9250;
alter bufferpool ITM DEFERRED size 3000;
alter bufferpool HST1 DEFERRED size 1500;
alter bufferpool HST2 DEFERRED size 1500;
alter bufferpool HST3 DEFERRED size 1500;
alter bufferpool HST4 DEFERRED size 1500;
alter bufferpool HST5 DEFERRED size 1500;
alter bufferpool HST6 DEFERRED size 1500;
alter bufferpool HST7 DEFERRED size 1500;
alter bufferpool HST8 DEFERRED size 1500;
alter bufferpool NEW1 DEFERRED size 960000;
alter bufferpool NEW2 DEFERRED size 960000;
alter bufferpool NEW3 DEFERRED size 960000;
alter bufferpool NEW4 DEFERRED size 960000;
alter bufferpool NEW5 DEFERRED size 960000;
alter bufferpool NEW6 DEFERRED size 960000;
alter bufferpool NEW7 DEFERRED size 960000;
alter bufferpool NEW8 DEFERRED size 960000;
alter bufferpool ORD1 DEFERRED size 700000;
alter bufferpool ORD2 DEFERRED size 700000;
alter bufferpool ORD3 DEFERRED size 700000;
alter bufferpool ORD4 DEFERRED size 700000;
alter bufferpool ORD5 DEFERRED size 700000;
alter bufferpool ORD6 DEFERRED size 700000;
alter bufferpool ORD7 DEFERRED size 700000;
alter bufferpool ORD8 DEFERRED size 700000;
alter bufferpool ORD_11 DEFERRED size 2200000;
alter bufferpool ORD_12 DEFERRED size 2200000;
alter bufferpool ORD_13 DEFERRED size 2200000;
alter bufferpool ORD_14 DEFERRED size 2200000;
alter bufferpool ORD_15 DEFERRED size 2200000;
alter bufferpool ORD_16 DEFERRED size 2200000;
alter bufferpool ORD_17 DEFERRED size 2200000;
alter bufferpool ORD_18 DEFERRED size 2200000;
alter bufferpool OLN1 DEFERRED size 1300000;
alter bufferpool OLN2 DEFERRED size 1300000;
alter bufferpool OLN3 DEFERRED size 1300000;
alter bufferpool OLN4 DEFERRED size 1300000;
alter bufferpool OLN5 DEFERRED size 1300000;
alter bufferpool OLN6 DEFERRED size 1300000;
alter bufferpool OLN7 DEFERRED size 1300000;

```

```

alter bufferpool OLN8 DEFERRED size 1300000;
alter bufferpool CST1 DEFERRED size 2900000;
alter bufferpool CST2 DEFERRED size 2900000;
alter bufferpool CST3 DEFERRED size 2900000;
alter bufferpool CST4 DEFERRED size 2900000;
alter bufferpool CST5 DEFERRED size 2900000;
alter bufferpool CST6 DEFERRED size 2900000;
alter bufferpool CST7 DEFERRED size 2900000;
alter bufferpool CST8 DEFERRED size 2900000;
alter bufferpool CST_I1 DEFERRED size 1300000;
alter bufferpool CST_I2 DEFERRED size 1300000;
alter bufferpool CST_I3 DEFERRED size 1300000;
alter bufferpool CST_I4 DEFERRED size 1300000;
alter bufferpool CST_I5 DEFERRED size 1300000;
alter bufferpool CST_I6 DEFERRED size 1300000;
alter bufferpool CST_I7 DEFERRED size 1300000;
alter bufferpool CST_I8 DEFERRED size 1300000;
alter bufferpool STK1 DEFERRED size 46000000;
alter bufferpool STK2 DEFERRED size 46000000;
alter bufferpool STK3 DEFERRED size 46000000;
alter bufferpool STK4 DEFERRED size 46000000;
alter bufferpool STK5 DEFERRED size 46000000;
alter bufferpool STK6 DEFERRED size 46000000;
alter bufferpool STK7 DEFERRED size 46000000;
alter bufferpool STK8 DEFERRED size 46000000;
connect reset;
terminate;

```

bp/alter_tablespace.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
-- 2004
-- All Rights Reserved.

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

-- Set Bufferpools For Tablespaces
connect to tpcc;

-- Non-Affinitized Bufferpools

alter tablespace TS_ITEM_001 bufferpool ITM;

-- Affinitized Bufferpools

-- WAREHOUSE

```

alter tablespace TS_WARE_001 bufferpool WAR1;
alter tablespace TS_WARE_002 bufferpool WAR2;
alter tablespace TS_WARE_003 bufferpool WAR3;
alter tablespace TS_WARE_004 bufferpool WAR4;
alter tablespace TS_WARE_005 bufferpool WAR5;
alter tablespace TS_WARE_006 bufferpool WAR6;
alter tablespace TS_WARE_007 bufferpool WAR7;

```

```

alter tablespace TS_WARE_008 bufferpool WAR8;
-- DISTRICT
alter tablespace TS_DIST_001 bufferpool DIS1;
alter tablespace TS_DIST_002 bufferpool DIS2;
alter tablespace TS_DIST_003 bufferpool DIS3;
alter tablespace TS_DIST_004 bufferpool DIS4;
alter tablespace TS_DIST_005 bufferpool DIS5;
alter tablespace TS_DIST_006 bufferpool DIS6;
alter tablespace TS_DIST_007 bufferpool DIS7;
alter tablespace TS_DIST_008 bufferpool DIS8;

```

-- HISTORY

```

alter tablespace TS_HISTORY_001 bufferpool HST1;
alter tablespace TS_HISTORY_002 bufferpool HST1;
alter tablespace TS_HISTORY_003 bufferpool HST2;
alter tablespace TS_HISTORY_004 bufferpool HST2;
alter tablespace TS_HISTORY_005 bufferpool HST3;
alter tablespace TS_HISTORY_006 bufferpool HST3;
alter tablespace TS_HISTORY_007 bufferpool HST4;
alter tablespace TS_HISTORY_008 bufferpool HST4;
alter tablespace TS_HISTORY_009 bufferpool HST5;
alter tablespace TS_HISTORY_010 bufferpool HST5;
alter tablespace TS_HISTORY_011 bufferpool HST6;
alter tablespace TS_HISTORY_012 bufferpool HST6;
alter tablespace TS_HISTORY_013 bufferpool HST7;
alter tablespace TS_HISTORY_014 bufferpool HST7;
alter tablespace TS_HISTORY_015 bufferpool HST8;
alter tablespace TS_HISTORY_016 bufferpool HST8;

```

-- NEWORDERA

```

alter tablespace TS_NEWORDA_001 bufferpool NEW1;
alter tablespace TS_NEWORDA_002 bufferpool NEW2;
alter tablespace TS_NEWORDA_003 bufferpool NEW3;
alter tablespace TS_NEWORDA_004 bufferpool NEW4;
alter tablespace TS_NEWORDA_005 bufferpool NEW5;
alter tablespace TS_NEWORDA_006 bufferpool NEW6;
alter tablespace TS_NEWORDA_007 bufferpool NEW7;
alter tablespace TS_NEWORDA_008 bufferpool NEW8;

```

-- NEWORDERB

```

alter tablespace TS_NEWORDB_001 bufferpool NEW1;
alter tablespace TS_NEWORDB_002 bufferpool NEW2;
alter tablespace TS_NEWORDB_003 bufferpool NEW3;
alter tablespace TS_NEWORDB_004 bufferpool NEW4;
alter tablespace TS_NEWORDB_005 bufferpool NEW5;
alter tablespace TS_NEWORDB_006 bufferpool NEW6;
alter tablespace TS_NEWORDB_007 bufferpool NEW7;
alter tablespace TS_NEWORDB_008 bufferpool NEW8;

```

-- ORDERS -- Partition 1

```

alter tablespace TS_ORDER_001 bufferpool ORD1;
alter tablespace TS_ORDER_002 bufferpool ORD1;
alter tablespace TS_ORDER_003 bufferpool ORD1;
alter tablespace TS_ORDER_004 bufferpool ORD1;
alter tablespace TS_ORDER_005 bufferpool ORD1;
alter tablespace TS_ORDER_006 bufferpool ORD1;
alter tablespace TS_ORDER_007 bufferpool ORD1;
alter tablespace TS_ORDER_008 bufferpool ORD1;
alter tablespace TS_ORDER_009 bufferpool ORD1;
alter tablespace TS_ORDER_010 bufferpool ORD1;
alter tablespace TS_ORDER_011 bufferpool ORD1;
alter tablespace TS_ORDER_012 bufferpool ORD1;
alter tablespace TS_ORDER_013 bufferpool ORD1;
alter tablespace TS_ORDER_014 bufferpool ORD1;
alter tablespace TS_ORDER_015 bufferpool ORD1;
alter tablespace TS_ORDER_016 bufferpool ORD1;

```

```

alter tablespace TS_ORDER_017 bufferpool ORD1;
alter tablespace TS_ORDER_018 bufferpool ORD1;
alter tablespace TS_ORDER_019 bufferpool ORD1;
alter tablespace TS_ORDER_020 bufferpool ORD1;
-- ORDERS -- Partition 2

```

```

alter tablespace TS_ORDER_021 bufferpool ORD2;
alter tablespace TS_ORDER_022 bufferpool ORD2;
alter tablespace TS_ORDER_023 bufferpool ORD2;
alter tablespace TS_ORDER_024 bufferpool ORD2;
alter tablespace TS_ORDER_025 bufferpool ORD2;
alter tablespace TS_ORDER_026 bufferpool ORD2;
alter tablespace TS_ORDER_027 bufferpool ORD2;
alter tablespace TS_ORDER_028 bufferpool ORD2;
alter tablespace TS_ORDER_029 bufferpool ORD2;
alter tablespace TS_ORDER_030 bufferpool ORD2;
alter tablespace TS_ORDER_031 bufferpool ORD2;
alter tablespace TS_ORDER_032 bufferpool ORD2;
alter tablespace TS_ORDER_033 bufferpool ORD2;
alter tablespace TS_ORDER_034 bufferpool ORD2;
alter tablespace TS_ORDER_035 bufferpool ORD2;
alter tablespace TS_ORDER_036 bufferpool ORD2;
alter tablespace TS_ORDER_037 bufferpool ORD2;
alter tablespace TS_ORDER_038 bufferpool ORD2;
alter tablespace TS_ORDER_039 bufferpool ORD2;
alter tablespace TS_ORDER_040 bufferpool ORD2;

```

-- ORDERS -- Partition 3

```

alter tablespace TS_ORDER_041 bufferpool ORD3;
alter tablespace TS_ORDER_042 bufferpool ORD3;
alter tablespace TS_ORDER_043 bufferpool ORD3;
alter tablespace TS_ORDER_044 bufferpool ORD3;
alter tablespace TS_ORDER_045 bufferpool ORD3;
alter tablespace TS_ORDER_046 bufferpool ORD3;
alter tablespace TS_ORDER_047 bufferpool ORD3;
alter tablespace TS_ORDER_048 bufferpool ORD3;
alter tablespace TS_ORDER_049 bufferpool ORD3;
alter tablespace TS_ORDER_050 bufferpool ORD3;
alter tablespace TS_ORDER_051 bufferpool ORD3;
alter tablespace TS_ORDER_052 bufferpool ORD3;
alter tablespace TS_ORDER_053 bufferpool ORD3;
alter tablespace TS_ORDER_054 bufferpool ORD3;
alter tablespace TS_ORDER_055 bufferpool ORD3;
alter tablespace TS_ORDER_056 bufferpool ORD3;
alter tablespace TS_ORDER_057 bufferpool ORD3;
alter tablespace TS_ORDER_058 bufferpool ORD3;
alter tablespace TS_ORDER_059 bufferpool ORD3;
alter tablespace TS_ORDER_060 bufferpool ORD3;

```

-- ORDERS -- Partition 4

```

alter tablespace TS_ORDER_061 bufferpool ORD4;
alter tablespace TS_ORDER_062 bufferpool ORD4;
alter tablespace TS_ORDER_063 bufferpool ORD4;
alter tablespace TS_ORDER_064 bufferpool ORD4;
alter tablespace TS_ORDER_065 bufferpool ORD4;
alter tablespace TS_ORDER_066 bufferpool ORD4;
alter tablespace TS_ORDER_067 bufferpool ORD4;
alter tablespace TS_ORDER_068 bufferpool ORD4;
alter tablespace TS_ORDER_069 bufferpool ORD4;
alter tablespace TS_ORDER_070 bufferpool ORD4;
alter tablespace TS_ORDER_071 bufferpool ORD4;
alter tablespace TS_ORDER_072 bufferpool ORD4;
alter tablespace TS_ORDER_073 bufferpool ORD4;
alter tablespace TS_ORDER_074 bufferpool ORD4;
alter tablespace TS_ORDER_075 bufferpool ORD4;

```


terminate;

bp/create_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  
-- 2004  
-- All Rights Reserved.  
--  
-- US Government Users Restricted Rights - Use, duplication or  
-- disclosure restricted by GSA ADP Schedule Contract with  
-- IBM Corp.  
-----  
-- Create Bufferpools  
connect to tpcc;  
create bufferpool IBMDEFAULTBP8K size 100 pagesize 8K;  
create bufferpool IBMDEFAULTBP16K size 100 pagesize 16K;  
create bufferpool WAR1 size 100;  
create bufferpool WAR2 size 100;  
create bufferpool WAR3 size 100;  
create bufferpool WAR4 size 100;  
create bufferpool WAR5 size 100;  
create bufferpool WAR6 size 100;  
create bufferpool WAR7 size 100;  
create bufferpool WAR8 size 100;  
create bufferpool DIS1 size 100;  
create bufferpool DIS2 size 100;  
create bufferpool DIS3 size 100;  
create bufferpool DIS4 size 100;  
create bufferpool DIS5 size 100;  
create bufferpool DIS6 size 100;  
create bufferpool DIS7 size 100;  
create bufferpool DIS8 size 100;  
create bufferpool ITM size 100;  
create bufferpool HST1 size 100 pagesize 16K;  
create bufferpool HST2 size 100 pagesize 16K;  
create bufferpool HST3 size 100 pagesize 16K;  
create bufferpool HST4 size 100 pagesize 16K;  
create bufferpool HST5 size 100 pagesize 16K;  
create bufferpool HST6 size 100 pagesize 16K;  
create bufferpool HST7 size 100 pagesize 16K;  
create bufferpool HST8 size 100 pagesize 16K;  
create bufferpool NEW1 size 100;  
create bufferpool NEW2 size 100;  
create bufferpool NEW3 size 100;  
create bufferpool NEW4 size 100;  
create bufferpool NEW5 size 100;  
create bufferpool NEW6 size 100;  
create bufferpool NEW7 size 100;  
create bufferpool NEW8 size 100;  
create bufferpool ORD1 size 100 pagesize 8K;  
create bufferpool ORD2 size 100 pagesize 8K;  
create bufferpool ORD3 size 100 pagesize 8K;  
create bufferpool ORD4 size 100 pagesize 8K;  
create bufferpool ORD5 size 100 pagesize 8K;  
create bufferpool ORD6 size 100 pagesize 8K;
```

```
create bufferpool ORD7 size 100 pagesize 8K;  
create bufferpool ORD8 size 100 pagesize 8K;  
create bufferpool ORD_I1 size 100 pagesize 8K;  
create bufferpool ORD_I2 size 100 pagesize 8K;  
create bufferpool ORD_I3 size 100 pagesize 8K;  
create bufferpool ORD_I4 size 100 pagesize 8K;  
create bufferpool ORD_I5 size 100 pagesize 8K;  
create bufferpool ORD_I6 size 100 pagesize 8K;  
create bufferpool ORD_I7 size 100 pagesize 8K;  
create bufferpool ORD_I8 size 100 pagesize 8K;  
create bufferpool OLN1 size 100 pagesize 8K;  
create bufferpool OLN2 size 100 pagesize 8K;  
create bufferpool OLN3 size 100 pagesize 8K;  
create bufferpool OLN4 size 100 pagesize 8K;  
create bufferpool OLN5 size 100 pagesize 8K;  
create bufferpool OLN6 size 100 pagesize 8K;  
create bufferpool OLN7 size 100 pagesize 8K;  
create bufferpool OLN8 size 100 pagesize 8K;  
create bufferpool CST1 size 100;  
create bufferpool CST2 size 100;  
create bufferpool CST3 size 100;  
create bufferpool CST4 size 100;  
create bufferpool CST5 size 100;  
create bufferpool CST6 size 100;  
create bufferpool CST7 size 100;  
create bufferpool CST8 size 100;  
create bufferpool CST_I1 size 100 pagesize 8K;  
create bufferpool CST_I2 size 100 pagesize 8K;  
create bufferpool CST_I3 size 100 pagesize 8K;  
create bufferpool CST_I4 size 100 pagesize 8K;  
create bufferpool CST_I5 size 100 pagesize 8K;  
create bufferpool CST_I6 size 100 pagesize 8K;  
create bufferpool CST_I7 size 100 pagesize 8K;  
create bufferpool CST_I8 size 100 pagesize 8K;  
create bufferpool STK1 size 100;  
create bufferpool STK2 size 100;  
create bufferpool STK3 size 100;  
create bufferpool STK4 size 100;  
create bufferpool STK5 size 100;  
create bufferpool STK6 size 100;  
create bufferpool STK7 size 100;  
create bufferpool STK8 size 100;  
connect reset;  
terminate;
```

ALTTBSP_PF_0.ddl

connect to TPCC;

```
connect to TPCC;  
alter tablespace is_customer_001 prefetchsize 0;  
alter tablespace is_customer_002 prefetchsize 0;  
alter tablespace is_customer_003 prefetchsize 0;  
alter tablespace is_customer_004 prefetchsize 0;  
alter tablespace is_customer_005 prefetchsize 0;  
alter tablespace is_customer_006 prefetchsize 0;  
alter tablespace is_customer_007 prefetchsize 0;  
alter tablespace is_customer_008 prefetchsize 0;  
alter tablespace is_customer_009 prefetchsize 0;  
alter tablespace is_customer_010 prefetchsize 0;
```

```
alter tablespace is_customer_011 prefetchsize 0;  
alter tablespace is_customer_012 prefetchsize 0;  
alter tablespace is_customer_013 prefetchsize 0;  
alter tablespace is_customer_014 prefetchsize 0;  
alter tablespace is_customer_015 prefetchsize 0;  
alter tablespace is_customer_016 prefetchsize 0;  
alter tablespace is_customer_017 prefetchsize 0;  
alter tablespace is_customer_018 prefetchsize 0;  
alter tablespace is_customer_019 prefetchsize 0;  
alter tablespace is_customer_020 prefetchsize 0;  
alter tablespace is_customer_021 prefetchsize 0;  
alter tablespace is_customer_022 prefetchsize 0;  
alter tablespace is_customer_023 prefetchsize 0;  
alter tablespace is_customer_024 prefetchsize 0;  
alter tablespace is_customer_025 prefetchsize 0;  
alter tablespace is_customer_026 prefetchsize 0;  
alter tablespace is_customer_027 prefetchsize 0;  
alter tablespace is_customer_028 prefetchsize 0;  
alter tablespace is_customer_029 prefetchsize 0;  
alter tablespace is_customer_030 prefetchsize 0;  
alter tablespace is_customer_031 prefetchsize 0;  
alter tablespace is_customer_032 prefetchsize 0;  
alter tablespace is_customer_033 prefetchsize 0;  
alter tablespace is_customer_034 prefetchsize 0;  
alter tablespace is_customer_035 prefetchsize 0;  
alter tablespace is_customer_036 prefetchsize 0;  
alter tablespace is_customer_037 prefetchsize 0;  
alter tablespace is_customer_038 prefetchsize 0;  
alter tablespace is_customer_039 prefetchsize 0;  
alter tablespace is_customer_040 prefetchsize 0;  
alter tablespace is_customer_041 prefetchsize 0;  
alter tablespace is_customer_042 prefetchsize 0;  
alter tablespace is_customer_043 prefetchsize 0;  
alter tablespace is_customer_044 prefetchsize 0;  
alter tablespace is_customer_045 prefetchsize 0;  
alter tablespace is_customer_046 prefetchsize 0;  
alter tablespace is_customer_047 prefetchsize 0;  
alter tablespace is_customer_048 prefetchsize 0;  
alter tablespace is_customer_049 prefetchsize 0;  
alter tablespace is_customer_050 prefetchsize 0;  
alter tablespace is_customer_051 prefetchsize 0;  
alter tablespace is_customer_052 prefetchsize 0;  
alter tablespace is_customer_053 prefetchsize 0;  
alter tablespace is_customer_054 prefetchsize 0;  
alter tablespace is_customer_055 prefetchsize 0;  
alter tablespace is_customer_056 prefetchsize 0;  
alter tablespace is_customer_057 prefetchsize 0;  
alter tablespace is_customer_058 prefetchsize 0;  
alter tablespace is_customer_059 prefetchsize 0;  
alter tablespace is_customer_060 prefetchsize 0;  
alter tablespace is_customer_061 prefetchsize 0;  
alter tablespace is_customer_062 prefetchsize 0;  
alter tablespace is_customer_063 prefetchsize 0;  
alter tablespace is_customer_064 prefetchsize 0;  
alter tablespace is_customer_065 prefetchsize 0;  
alter tablespace is_customer_066 prefetchsize 0;  
alter tablespace is_customer_067 prefetchsize 0;  
alter tablespace is_customer_068 prefetchsize 0;  
alter tablespace is_customer_069 prefetchsize 0;  
alter tablespace is_customer_070 prefetchsize 0;  
alter tablespace is_customer_071 prefetchsize 0;  
alter tablespace is_customer_072 prefetchsize 0;
```



```

ALTER TABLE CUSTOMER9 ADD CONSTRAINT
CUSTOMER9CKC CHECK (C_W_ID BETWEEN 12801 AND
14400);
SET INTEGRITY FOR CUSTOMER9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER10 OFF;
ALTER TABLE CUSTOMER10 DROP CONSTRAINT
CUSTOMER10CKC;
ALTER TABLE CUSTOMER10 ADD CONSTRAINT
CUSTOMER10CKC CHECK (C_W_ID BETWEEN 14401 AND
16000);
SET INTEGRITY FOR CUSTOMER10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER11 OFF;
ALTER TABLE CUSTOMER11 DROP CONSTRAINT
CUSTOMER11CKC;
ALTER TABLE CUSTOMER11 ADD CONSTRAINT
CUSTOMER11CKC CHECK (C_W_ID BETWEEN 16001 AND
17600);
SET INTEGRITY FOR CUSTOMER11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER12 OFF;
ALTER TABLE CUSTOMER12 DROP CONSTRAINT
CUSTOMER12CKC;
ALTER TABLE CUSTOMER12 ADD CONSTRAINT
CUSTOMER12CKC CHECK (C_W_ID BETWEEN 17601 AND
19200);
SET INTEGRITY FOR CUSTOMER12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER13 OFF;
ALTER TABLE CUSTOMER13 DROP CONSTRAINT
CUSTOMER13CKC;
ALTER TABLE CUSTOMER13 ADD CONSTRAINT
CUSTOMER13CKC CHECK (C_W_ID BETWEEN 19201 AND
20800);
SET INTEGRITY FOR CUSTOMER13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER14 OFF;
ALTER TABLE CUSTOMER14 DROP CONSTRAINT
CUSTOMER14CKC;
ALTER TABLE CUSTOMER14 ADD CONSTRAINT
CUSTOMER14CKC CHECK (C_W_ID BETWEEN 20801 AND
22400);
SET INTEGRITY FOR CUSTOMER14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER15 OFF;
ALTER TABLE CUSTOMER15 DROP CONSTRAINT
CUSTOMER15CKC;

```

```

ALTER TABLE CUSTOMER15 ADD CONSTRAINT
CUSTOMER15CKC CHECK (C_W_ID BETWEEN 22401 AND
24000);
SET INTEGRITY FOR CUSTOMER15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER16 OFF;
ALTER TABLE CUSTOMER16 DROP CONSTRAINT
CUSTOMER16CKC;
ALTER TABLE CUSTOMER16 ADD CONSTRAINT
CUSTOMER16CKC CHECK (C_W_ID BETWEEN 24001 AND
25600);
SET INTEGRITY FOR CUSTOMER16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER17 OFF;
ALTER TABLE CUSTOMER17 DROP CONSTRAINT
CUSTOMER17CKC;
ALTER TABLE CUSTOMER17 ADD CONSTRAINT
CUSTOMER17CKC CHECK (C_W_ID BETWEEN 25601 AND
27200);
SET INTEGRITY FOR CUSTOMER17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER18 OFF;
ALTER TABLE CUSTOMER18 DROP CONSTRAINT
CUSTOMER18CKC;
ALTER TABLE CUSTOMER18 ADD CONSTRAINT
CUSTOMER18CKC CHECK (C_W_ID BETWEEN 27201 AND
28800);
SET INTEGRITY FOR CUSTOMER18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER19 OFF;
ALTER TABLE CUSTOMER19 DROP CONSTRAINT
CUSTOMER19CKC;
ALTER TABLE CUSTOMER19 ADD CONSTRAINT
CUSTOMER19CKC CHECK (C_W_ID BETWEEN 28801 AND
30400);
SET INTEGRITY FOR CUSTOMER19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER20 OFF;
ALTER TABLE CUSTOMER20 DROP CONSTRAINT
CUSTOMER20CKC;
ALTER TABLE CUSTOMER20 ADD CONSTRAINT
CUSTOMER20CKC CHECK (C_W_ID BETWEEN 30401 AND
32000);
SET INTEGRITY FOR CUSTOMER20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER21 OFF;
ALTER TABLE CUSTOMER21 DROP CONSTRAINT
CUSTOMER21CKC;

```

```

ALTER TABLE CUSTOMER21 ADD CONSTRAINT
CUSTOMER21CKC CHECK (C_W_ID BETWEEN 32001 AND
33600);
SET INTEGRITY FOR CUSTOMER21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER22 OFF;
ALTER TABLE CUSTOMER22 DROP CONSTRAINT
CUSTOMER22CKC;
ALTER TABLE CUSTOMER22 ADD CONSTRAINT
CUSTOMER22CKC CHECK (C_W_ID BETWEEN 33601 AND
35200);
SET INTEGRITY FOR CUSTOMER22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER23 OFF;
ALTER TABLE CUSTOMER23 DROP CONSTRAINT
CUSTOMER23CKC;
ALTER TABLE CUSTOMER23 ADD CONSTRAINT
CUSTOMER23CKC CHECK (C_W_ID BETWEEN 35201 AND
36800);
SET INTEGRITY FOR CUSTOMER23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER24 OFF;
ALTER TABLE CUSTOMER24 DROP CONSTRAINT
CUSTOMER24CKC;
ALTER TABLE CUSTOMER24 ADD CONSTRAINT
CUSTOMER24CKC CHECK (C_W_ID BETWEEN 36801 AND
38400);
SET INTEGRITY FOR CUSTOMER24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER25 OFF;
ALTER TABLE CUSTOMER25 DROP CONSTRAINT
CUSTOMER25CKC;
ALTER TABLE CUSTOMER25 ADD CONSTRAINT
CUSTOMER25CKC CHECK (C_W_ID BETWEEN 38401 AND
40000);
SET INTEGRITY FOR CUSTOMER25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER26 OFF;
ALTER TABLE CUSTOMER26 DROP CONSTRAINT
CUSTOMER26CKC;
ALTER TABLE CUSTOMER26 ADD CONSTRAINT
CUSTOMER26CKC CHECK (C_W_ID BETWEEN 40001 AND
41600);
SET INTEGRITY FOR CUSTOMER26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER27 OFF;
ALTER TABLE CUSTOMER27 DROP CONSTRAINT
CUSTOMER27CKC;

```



```

ALTER TABLE CUSTOMER27 ADD CONSTRAINT
CUSTOMER27CKC CHECK (C_W_ID BETWEEN 41601 AND
43200);
SET INTEGRITY FOR CUSTOMER27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER28 OFF;
ALTER TABLE CUSTOMER28 DROP CONSTRAINT
CUSTOMER28CKC;
ALTER TABLE CUSTOMER28 ADD CONSTRAINT
CUSTOMER28CKC CHECK (C_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER29 OFF;
ALTER TABLE CUSTOMER29 DROP CONSTRAINT
CUSTOMER29CKC;
ALTER TABLE CUSTOMER29 ADD CONSTRAINT
CUSTOMER29CKC CHECK (C_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR CUSTOMER29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER30 OFF;
ALTER TABLE CUSTOMER30 DROP CONSTRAINT
CUSTOMER30CKC;
ALTER TABLE CUSTOMER30 ADD CONSTRAINT
CUSTOMER30CKC CHECK (C_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR CUSTOMER30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER31 OFF;
ALTER TABLE CUSTOMER31 DROP CONSTRAINT
CUSTOMER31CKC;
ALTER TABLE CUSTOMER31 ADD CONSTRAINT
CUSTOMER31CKC CHECK (C_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR CUSTOMER31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER32 OFF;
ALTER TABLE CUSTOMER32 DROP CONSTRAINT
CUSTOMER32CKC;
ALTER TABLE CUSTOMER32 ADD CONSTRAINT
CUSTOMER32CKC CHECK (C_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER33 OFF;
ALTER TABLE CUSTOMER33 DROP CONSTRAINT
CUSTOMER33CKC;

```

```

ALTER TABLE CUSTOMER33 ADD CONSTRAINT
CUSTOMER33CKC CHECK (C_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR CUSTOMER33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER34 OFF;
ALTER TABLE CUSTOMER34 DROP CONSTRAINT
CUSTOMER34CKC;
ALTER TABLE CUSTOMER34 ADD CONSTRAINT
CUSTOMER34CKC CHECK (C_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR CUSTOMER34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER35 OFF;
ALTER TABLE CUSTOMER35 DROP CONSTRAINT
CUSTOMER35CKC;
ALTER TABLE CUSTOMER35 ADD CONSTRAINT
CUSTOMER35CKC CHECK (C_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR CUSTOMER35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER36 OFF;
ALTER TABLE CUSTOMER36 DROP CONSTRAINT
CUSTOMER36CKC;
ALTER TABLE CUSTOMER36 ADD CONSTRAINT
CUSTOMER36CKC CHECK (C_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR CUSTOMER36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER37 OFF;
ALTER TABLE CUSTOMER37 DROP CONSTRAINT
CUSTOMER37CKC;
ALTER TABLE CUSTOMER37 ADD CONSTRAINT
CUSTOMER37CKC CHECK (C_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR CUSTOMER37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER38 OFF;
ALTER TABLE CUSTOMER38 DROP CONSTRAINT
CUSTOMER38CKC;
ALTER TABLE CUSTOMER38 ADD CONSTRAINT
CUSTOMER38CKC CHECK (C_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR CUSTOMER38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER39 OFF;
ALTER TABLE CUSTOMER39 DROP CONSTRAINT
CUSTOMER39CKC;

```

```

ALTER TABLE CUSTOMER39 ADD CONSTRAINT
CUSTOMER39CKC CHECK (C_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR CUSTOMER39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER40 OFF;
ALTER TABLE CUSTOMER40 DROP CONSTRAINT
CUSTOMER40CKC;
ALTER TABLE CUSTOMER40 ADD CONSTRAINT
CUSTOMER40CKC CHECK (C_W_ID BETWEEN 62401 AND
64000);
SET INTEGRITY FOR CUSTOMER40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER41 OFF;
ALTER TABLE CUSTOMER41 DROP CONSTRAINT
CUSTOMER41CKC;
ALTER TABLE CUSTOMER41 ADD CONSTRAINT
CUSTOMER41CKC CHECK (C_W_ID BETWEEN 64001 AND
65600);
SET INTEGRITY FOR CUSTOMER41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER42 OFF;
ALTER TABLE CUSTOMER42 DROP CONSTRAINT
CUSTOMER42CKC;
ALTER TABLE CUSTOMER42 ADD CONSTRAINT
CUSTOMER42CKC CHECK (C_W_ID BETWEEN 65601 AND
67200);
SET INTEGRITY FOR CUSTOMER42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER43 OFF;
ALTER TABLE CUSTOMER43 DROP CONSTRAINT
CUSTOMER43CKC;
ALTER TABLE CUSTOMER43 ADD CONSTRAINT
CUSTOMER43CKC CHECK (C_W_ID BETWEEN 67201 AND
68800);
SET INTEGRITY FOR CUSTOMER43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER44 OFF;
ALTER TABLE CUSTOMER44 DROP CONSTRAINT
CUSTOMER44CKC;
ALTER TABLE CUSTOMER44 ADD CONSTRAINT
CUSTOMER44CKC CHECK (C_W_ID BETWEEN 68801 AND
70400);
SET INTEGRITY FOR CUSTOMER44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER45 OFF;
ALTER TABLE CUSTOMER45 DROP CONSTRAINT
CUSTOMER45CKC;

```

```

ALTER TABLE CUSTOMER45 ADD CONSTRAINT
CUSTOMER45CKC CHECK (C_W_ID BETWEEN 70401 AND
72000);
SET INTEGRITY FOR CUSTOMER45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER46 OFF;
ALTER TABLE CUSTOMER46 DROP CONSTRAINT
CUSTOMER46CKC;
ALTER TABLE CUSTOMER46 ADD CONSTRAINT
CUSTOMER46CKC CHECK (C_W_ID BETWEEN 72001 AND
73600);
SET INTEGRITY FOR CUSTOMER46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER47 OFF;
ALTER TABLE CUSTOMER47 DROP CONSTRAINT
CUSTOMER47CKC;
ALTER TABLE CUSTOMER47 ADD CONSTRAINT
CUSTOMER47CKC CHECK (C_W_ID BETWEEN 73601 AND
75200);
SET INTEGRITY FOR CUSTOMER47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER48 OFF;
ALTER TABLE CUSTOMER48 DROP CONSTRAINT
CUSTOMER48CKC;
ALTER TABLE CUSTOMER48 ADD CONSTRAINT
CUSTOMER48CKC CHECK (C_W_ID BETWEEN 75201 AND
76800);
SET INTEGRITY FOR CUSTOMER48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER49 OFF;
ALTER TABLE CUSTOMER49 DROP CONSTRAINT
CUSTOMER49CKC;
ALTER TABLE CUSTOMER49 ADD CONSTRAINT
CUSTOMER49CKC CHECK (C_W_ID BETWEEN 76801 AND
78400);
SET INTEGRITY FOR CUSTOMER49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER50 OFF;
ALTER TABLE CUSTOMER50 DROP CONSTRAINT
CUSTOMER50CKC;
ALTER TABLE CUSTOMER50 ADD CONSTRAINT
CUSTOMER50CKC CHECK (C_W_ID BETWEEN 78401 AND
80000);
SET INTEGRITY FOR CUSTOMER50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER51 OFF;
ALTER TABLE CUSTOMER51 DROP CONSTRAINT
CUSTOMER51CKC;

```

```

ALTER TABLE CUSTOMER51 ADD CONSTRAINT
CUSTOMER51CKC CHECK (C_W_ID BETWEEN 80001 AND
81600);
SET INTEGRITY FOR CUSTOMER51 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER52 OFF;
ALTER TABLE CUSTOMER52 DROP CONSTRAINT
CUSTOMER52CKC;
ALTER TABLE CUSTOMER52 ADD CONSTRAINT
CUSTOMER52CKC CHECK (C_W_ID BETWEEN 81601 AND
83200);
SET INTEGRITY FOR CUSTOMER52 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER53 OFF;
ALTER TABLE CUSTOMER53 DROP CONSTRAINT
CUSTOMER53CKC;
ALTER TABLE CUSTOMER53 ADD CONSTRAINT
CUSTOMER53CKC CHECK (C_W_ID BETWEEN 83201 AND
84800);
SET INTEGRITY FOR CUSTOMER53 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER54 OFF;
ALTER TABLE CUSTOMER54 DROP CONSTRAINT
CUSTOMER54CKC;
ALTER TABLE CUSTOMER54 ADD CONSTRAINT
CUSTOMER54CKC CHECK (C_W_ID BETWEEN 84801 AND
86400);
SET INTEGRITY FOR CUSTOMER54 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER55 OFF;
ALTER TABLE CUSTOMER55 DROP CONSTRAINT
CUSTOMER55CKC;
ALTER TABLE CUSTOMER55 ADD CONSTRAINT
CUSTOMER55CKC CHECK (C_W_ID BETWEEN 86401 AND
88000);
SET INTEGRITY FOR CUSTOMER55 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER56 OFF;
ALTER TABLE CUSTOMER56 DROP CONSTRAINT
CUSTOMER56CKC;
ALTER TABLE CUSTOMER56 ADD CONSTRAINT
CUSTOMER56CKC CHECK (C_W_ID BETWEEN 88001 AND
89600);
SET INTEGRITY FOR CUSTOMER56 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER57 OFF;
ALTER TABLE CUSTOMER57 DROP CONSTRAINT
CUSTOMER57CKC;

```

```

ALTER TABLE CUSTOMER57 ADD CONSTRAINT
CUSTOMER57CKC CHECK (C_W_ID BETWEEN 89601 AND
91200);
SET INTEGRITY FOR CUSTOMER57 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER58 OFF;
ALTER TABLE CUSTOMER58 DROP CONSTRAINT
CUSTOMER58CKC;
ALTER TABLE CUSTOMER58 ADD CONSTRAINT
CUSTOMER58CKC CHECK (C_W_ID BETWEEN 91201 AND
92800);
SET INTEGRITY FOR CUSTOMER58 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER59 OFF;
ALTER TABLE CUSTOMER59 DROP CONSTRAINT
CUSTOMER59CKC;
ALTER TABLE CUSTOMER59 ADD CONSTRAINT
CUSTOMER59CKC CHECK (C_W_ID BETWEEN 92801 AND
94400);
SET INTEGRITY FOR CUSTOMER59 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER60 OFF;
ALTER TABLE CUSTOMER60 DROP CONSTRAINT
CUSTOMER60CKC;
ALTER TABLE CUSTOMER60 ADD CONSTRAINT
CUSTOMER60CKC CHECK (C_W_ID BETWEEN 94401 AND
96000);
SET INTEGRITY FOR CUSTOMER60 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER61 OFF;
ALTER TABLE CUSTOMER61 DROP CONSTRAINT
CUSTOMER61CKC;
ALTER TABLE CUSTOMER61 ADD CONSTRAINT
CUSTOMER61CKC CHECK (C_W_ID BETWEEN 96001 AND
97600);
SET INTEGRITY FOR CUSTOMER61 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER62 OFF;
ALTER TABLE CUSTOMER62 DROP CONSTRAINT
CUSTOMER62CKC;
ALTER TABLE CUSTOMER62 ADD CONSTRAINT
CUSTOMER62CKC CHECK (C_W_ID BETWEEN 97601 AND
99200);
SET INTEGRITY FOR CUSTOMER62 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER63 OFF;
ALTER TABLE CUSTOMER63 DROP CONSTRAINT
CUSTOMER63CKC;

```

```

ALTER TABLE CUSTOMER63 ADD CONSTRAINT
CUSTOMER63CKC CHECK (C_W_ID BETWEEN 99201 AND
100800);
SET INTEGRITY FOR CUSTOMER63 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER64 OFF;
ALTER TABLE CUSTOMER64 DROP CONSTRAINT
CUSTOMER64CKC;
ALTER TABLE CUSTOMER64 ADD CONSTRAINT
CUSTOMER64CKC CHECK (C_W_ID BETWEEN 100801 AND
102400);
SET INTEGRITY FOR CUSTOMER64 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER65 OFF;
ALTER TABLE CUSTOMER65 DROP CONSTRAINT
CUSTOMER65CKC;
ALTER TABLE CUSTOMER65 ADD CONSTRAINT
CUSTOMER65CKC CHECK (C_W_ID BETWEEN 102401 AND
104000);
SET INTEGRITY FOR CUSTOMER65 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER66 OFF;
ALTER TABLE CUSTOMER66 DROP CONSTRAINT
CUSTOMER66CKC;
ALTER TABLE CUSTOMER66 ADD CONSTRAINT
CUSTOMER66CKC CHECK (C_W_ID BETWEEN 104001 AND
105600);
SET INTEGRITY FOR CUSTOMER66 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER67 OFF;
ALTER TABLE CUSTOMER67 DROP CONSTRAINT
CUSTOMER67CKC;
ALTER TABLE CUSTOMER67 ADD CONSTRAINT
CUSTOMER67CKC CHECK (C_W_ID BETWEEN 105601 AND
107200);
SET INTEGRITY FOR CUSTOMER67 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER68 OFF;
ALTER TABLE CUSTOMER68 DROP CONSTRAINT
CUSTOMER68CKC;
ALTER TABLE CUSTOMER68 ADD CONSTRAINT
CUSTOMER68CKC CHECK (C_W_ID BETWEEN 107201 AND
108800);
SET INTEGRITY FOR CUSTOMER68 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER69 OFF;
ALTER TABLE CUSTOMER69 DROP CONSTRAINT
CUSTOMER69CKC;

```

```

ALTER TABLE CUSTOMER69 ADD CONSTRAINT
CUSTOMER69CKC CHECK (C_W_ID BETWEEN 108801 AND
110400);
SET INTEGRITY FOR CUSTOMER69 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER70 OFF;
ALTER TABLE CUSTOMER70 DROP CONSTRAINT
CUSTOMER70CKC;
ALTER TABLE CUSTOMER70 ADD CONSTRAINT
CUSTOMER70CKC CHECK (C_W_ID BETWEEN 110401 AND
112000);
SET INTEGRITY FOR CUSTOMER70 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER71 OFF;
ALTER TABLE CUSTOMER71 DROP CONSTRAINT
CUSTOMER71CKC;
ALTER TABLE CUSTOMER71 ADD CONSTRAINT
CUSTOMER71CKC CHECK (C_W_ID BETWEEN 112001 AND
113600);
SET INTEGRITY FOR CUSTOMER71 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER72 OFF;
ALTER TABLE CUSTOMER72 DROP CONSTRAINT
CUSTOMER72CKC;
ALTER TABLE CUSTOMER72 ADD CONSTRAINT
CUSTOMER72CKC CHECK (C_W_ID BETWEEN 113601 AND
115200);
SET INTEGRITY FOR CUSTOMER72 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER73 OFF;
ALTER TABLE CUSTOMER73 DROP CONSTRAINT
CUSTOMER73CKC;
ALTER TABLE CUSTOMER73 ADD CONSTRAINT
CUSTOMER73CKC CHECK (C_W_ID BETWEEN 115201 AND
116800);
SET INTEGRITY FOR CUSTOMER73 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER74 OFF;
ALTER TABLE CUSTOMER74 DROP CONSTRAINT
CUSTOMER74CKC;
ALTER TABLE CUSTOMER74 ADD CONSTRAINT
CUSTOMER74CKC CHECK (C_W_ID BETWEEN 116801 AND
118400);
SET INTEGRITY FOR CUSTOMER74 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER75 OFF;
ALTER TABLE CUSTOMER75 DROP CONSTRAINT
CUSTOMER75CKC;

```

```

ALTER TABLE CUSTOMER75 ADD CONSTRAINT
CUSTOMER75CKC CHECK (C_W_ID BETWEEN 118401 AND
120000);
SET INTEGRITY FOR CUSTOMER75 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER76 OFF;
ALTER TABLE CUSTOMER76 DROP CONSTRAINT
CUSTOMER76CKC;
ALTER TABLE CUSTOMER76 ADD CONSTRAINT
CUSTOMER76CKC CHECK (C_W_ID BETWEEN 120001 AND
121600);
SET INTEGRITY FOR CUSTOMER76 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER77 OFF;
ALTER TABLE CUSTOMER77 DROP CONSTRAINT
CUSTOMER77CKC;
ALTER TABLE CUSTOMER77 ADD CONSTRAINT
CUSTOMER77CKC CHECK (C_W_ID BETWEEN 121601 AND
123200);
SET INTEGRITY FOR CUSTOMER77 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER78 OFF;
ALTER TABLE CUSTOMER78 DROP CONSTRAINT
CUSTOMER78CKC;
ALTER TABLE CUSTOMER78 ADD CONSTRAINT
CUSTOMER78CKC CHECK (C_W_ID BETWEEN 123201 AND
124800);
SET INTEGRITY FOR CUSTOMER78 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER79 OFF;
ALTER TABLE CUSTOMER79 DROP CONSTRAINT
CUSTOMER79CKC;
ALTER TABLE CUSTOMER79 ADD CONSTRAINT
CUSTOMER79CKC CHECK (C_W_ID BETWEEN 124801 AND
126400);
SET INTEGRITY FOR CUSTOMER79 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER80 OFF;
ALTER TABLE CUSTOMER80 DROP CONSTRAINT
CUSTOMER80CKC;
ALTER TABLE CUSTOMER80 ADD CONSTRAINT
CUSTOMER80CKC CHECK (C_W_ID BETWEEN 126401 AND
128000);
SET INTEGRITY FOR CUSTOMER80 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER81 OFF;
ALTER TABLE CUSTOMER81 DROP CONSTRAINT
CUSTOMER81CKC;

```

```

ALTER TABLE CUSTOMER81 ADD CONSTRAINT
CUSTOMER81CKC CHECK (C_W_ID BETWEEN 128001 AND
129600);
SET INTEGRITY FOR CUSTOMER81 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER82 OFF;
ALTER TABLE CUSTOMER82 DROP CONSTRAINT
CUSTOMER82CKC;
ALTER TABLE CUSTOMER82 ADD CONSTRAINT
CUSTOMER82CKC CHECK (C_W_ID BETWEEN 129601 AND
131200);
SET INTEGRITY FOR CUSTOMER82 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER83 OFF;
ALTER TABLE CUSTOMER83 DROP CONSTRAINT
CUSTOMER83CKC;
ALTER TABLE CUSTOMER83 ADD CONSTRAINT
CUSTOMER83CKC CHECK (C_W_ID BETWEEN 131201 AND
132800);
SET INTEGRITY FOR CUSTOMER83 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER84 OFF;
ALTER TABLE CUSTOMER84 DROP CONSTRAINT
CUSTOMER84CKC;
ALTER TABLE CUSTOMER84 ADD CONSTRAINT
CUSTOMER84CKC CHECK (C_W_ID BETWEEN 132801 AND
134400);
SET INTEGRITY FOR CUSTOMER84 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER85 OFF;
ALTER TABLE CUSTOMER85 DROP CONSTRAINT
CUSTOMER85CKC;
ALTER TABLE CUSTOMER85 ADD CONSTRAINT
CUSTOMER85CKC CHECK (C_W_ID BETWEEN 134401 AND
136000);
SET INTEGRITY FOR CUSTOMER85 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER86 OFF;
ALTER TABLE CUSTOMER86 DROP CONSTRAINT
CUSTOMER86CKC;
ALTER TABLE CUSTOMER86 ADD CONSTRAINT
CUSTOMER86CKC CHECK (C_W_ID BETWEEN 136001 AND
137600);
SET INTEGRITY FOR CUSTOMER86 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER87 OFF;
ALTER TABLE CUSTOMER87 DROP CONSTRAINT
CUSTOMER87CKC;

```

```

ALTER TABLE CUSTOMER87 ADD CONSTRAINT
CUSTOMER87CKC CHECK (C_W_ID BETWEEN 137601 AND
139200);
SET INTEGRITY FOR CUSTOMER87 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER88 OFF;
ALTER TABLE CUSTOMER88 DROP CONSTRAINT
CUSTOMER88CKC;
ALTER TABLE CUSTOMER88 ADD CONSTRAINT
CUSTOMER88CKC CHECK (C_W_ID BETWEEN 139201 AND
140800);
SET INTEGRITY FOR CUSTOMER88 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER89 OFF;
ALTER TABLE CUSTOMER89 DROP CONSTRAINT
CUSTOMER89CKC;
ALTER TABLE CUSTOMER89 ADD CONSTRAINT
CUSTOMER89CKC CHECK (C_W_ID BETWEEN 140801 AND
142400);
SET INTEGRITY FOR CUSTOMER89 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER90 OFF;
ALTER TABLE CUSTOMER90 DROP CONSTRAINT
CUSTOMER90CKC;
ALTER TABLE CUSTOMER90 ADD CONSTRAINT
CUSTOMER90CKC CHECK (C_W_ID BETWEEN 142401 AND
144000);
SET INTEGRITY FOR CUSTOMER90 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER91 OFF;
ALTER TABLE CUSTOMER91 DROP CONSTRAINT
CUSTOMER91CKC;
ALTER TABLE CUSTOMER91 ADD CONSTRAINT
CUSTOMER91CKC CHECK (C_W_ID BETWEEN 144001 AND
145600);
SET INTEGRITY FOR CUSTOMER91 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER92 OFF;
ALTER TABLE CUSTOMER92 DROP CONSTRAINT
CUSTOMER92CKC;
ALTER TABLE CUSTOMER92 ADD CONSTRAINT
CUSTOMER92CKC CHECK (C_W_ID BETWEEN 145601 AND
147200);
SET INTEGRITY FOR CUSTOMER92 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER93 OFF;
ALTER TABLE CUSTOMER93 DROP CONSTRAINT
CUSTOMER93CKC;

```

```

ALTER TABLE CUSTOMER93 ADD CONSTRAINT
CUSTOMER93CKC CHECK (C_W_ID BETWEEN 147201 AND
148800);
SET INTEGRITY FOR CUSTOMER93 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER94 OFF;
ALTER TABLE CUSTOMER94 DROP CONSTRAINT
CUSTOMER94CKC;
ALTER TABLE CUSTOMER94 ADD CONSTRAINT
CUSTOMER94CKC CHECK (C_W_ID BETWEEN 148801 AND
150400);
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 150401 AND
152000);
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 152001 AND
153600);
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 153601 AND
155200);
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 155201 AND
156800);
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 156801 AND
158400);
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 158401
AND 160000);
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 160001
AND 161600);
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 161601
AND 163200);
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 163201
AND 164800);
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 164801
AND 166400);
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 166401
AND 168000);
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 168001
AND 169600);
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 169601
AND 171200);
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 171201
AND 172800);
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 172801
AND 174400);
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 174401
AND 176000);
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 176001
AND 177600);
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 177601
AND 179200);
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 179201
AND 180800);
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 180801
AND 182400);
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 182401
AND 184000);
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 184001
AND 185600);
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 185601
AND 187200);
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 187201
AND 188800);
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 188801
AND 190400);
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 BETWEEN 190401
AND 192000);
SET INTEGRITY FOR CUSTOMER120 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER121 OFF;
ALTER TABLE CUSTOMER121 DROP CONSTRAINT
CUSTOMER121CKC;
ALTER TABLE CUSTOMER121 ADD CONSTRAINT
CUSTOMER121CKC CHECK (C_W_ID BETWEEN 192001
AND 193600);
SET INTEGRITY FOR CUSTOMER121 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER122 OFF;
ALTER TABLE CUSTOMER122 DROP CONSTRAINT
CUSTOMER122CKC;
ALTER TABLE CUSTOMER122 ADD CONSTRAINT
CUSTOMER122CKC CHECK (C_W_ID BETWEEN 193601
AND 195200);
SET INTEGRITY FOR CUSTOMER122 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER123 OFF;
ALTER TABLE CUSTOMER123 DROP CONSTRAINT
CUSTOMER123CKC;

```

```

ALTER TABLE CUSTOMER123 ADD CONSTRAINT
CUSTOMER123CKC CHECK (C_W_ID BETWEEN 195201
AND 196800);
SET INTEGRITY FOR CUSTOMER123 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER124 OFF;
ALTER TABLE CUSTOMER124 DROP CONSTRAINT
CUSTOMER124CKC;
ALTER TABLE CUSTOMER124 ADD CONSTRAINT
CUSTOMER124CKC CHECK (C_W_ID BETWEEN 196801
AND 198400);
SET INTEGRITY FOR CUSTOMER124 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER125 OFF;
ALTER TABLE CUSTOMER125 DROP CONSTRAINT
CUSTOMER125CKC;
ALTER TABLE CUSTOMER125 ADD CONSTRAINT
CUSTOMER125CKC CHECK (C_W_ID BETWEEN 198401
AND 200000);
SET INTEGRITY FOR CUSTOMER125 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER126 OFF;
ALTER TABLE CUSTOMER126 DROP CONSTRAINT
CUSTOMER126CKC;
ALTER TABLE CUSTOMER126 ADD CONSTRAINT
CUSTOMER126CKC CHECK (C_W_ID BETWEEN 200001
AND 201600);
SET INTEGRITY FOR CUSTOMER126 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER127 OFF;
ALTER TABLE CUSTOMER127 DROP CONSTRAINT
CUSTOMER127CKC;
ALTER TABLE CUSTOMER127 ADD CONSTRAINT
CUSTOMER127CKC CHECK (C_W_ID BETWEEN 201601
AND 203200);
SET INTEGRITY FOR CUSTOMER127 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER128 OFF;
ALTER TABLE CUSTOMER128 DROP CONSTRAINT
CUSTOMER128CKC;
ALTER TABLE CUSTOMER128 ADD CONSTRAINT
CUSTOMER128CKC CHECK (C_W_ID BETWEEN 203201
AND 204800);
SET INTEGRITY FOR CUSTOMER128 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER129 OFF;
ALTER TABLE CUSTOMER129 DROP CONSTRAINT
CUSTOMER129CKC;

```

```

ALTER TABLE CUSTOMER129 ADD CONSTRAINT
CUSTOMER129CKC CHECK (C_W_ID BETWEEN 204801
AND 206400);
SET INTEGRITY FOR CUSTOMER129 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER130 OFF;
ALTER TABLE CUSTOMER130 DROP CONSTRAINT
CUSTOMER130CKC;
ALTER TABLE CUSTOMER130 ADD CONSTRAINT
CUSTOMER130CKC CHECK (C_W_ID BETWEEN 206401
AND 208000);
SET INTEGRITY FOR CUSTOMER130 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER131 OFF;
ALTER TABLE CUSTOMER131 DROP CONSTRAINT
CUSTOMER131CKC;
ALTER TABLE CUSTOMER131 ADD CONSTRAINT
CUSTOMER131CKC CHECK (C_W_ID BETWEEN 208001
AND 209600);
SET INTEGRITY FOR CUSTOMER131 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER132 OFF;
ALTER TABLE CUSTOMER132 DROP CONSTRAINT
CUSTOMER132CKC;
ALTER TABLE CUSTOMER132 ADD CONSTRAINT
CUSTOMER132CKC CHECK (C_W_ID BETWEEN 209601
AND 211200);
SET INTEGRITY FOR CUSTOMER132 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER133 OFF;
ALTER TABLE CUSTOMER133 DROP CONSTRAINT
CUSTOMER133CKC;
ALTER TABLE CUSTOMER133 ADD CONSTRAINT
CUSTOMER133CKC CHECK (C_W_ID BETWEEN 211201
AND 212800);
SET INTEGRITY FOR CUSTOMER133 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER134 OFF;
ALTER TABLE CUSTOMER134 DROP CONSTRAINT
CUSTOMER134CKC;
ALTER TABLE CUSTOMER134 ADD CONSTRAINT
CUSTOMER134CKC CHECK (C_W_ID BETWEEN 212801
AND 214400);
SET INTEGRITY FOR CUSTOMER134 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER135 OFF;
ALTER TABLE CUSTOMER135 DROP CONSTRAINT
CUSTOMER135CKC;

```

```

ALTER TABLE CUSTOMER135 ADD CONSTRAINT
CUSTOMER135CKC CHECK (C_W_ID BETWEEN 214401
AND 216000);
SET INTEGRITY FOR CUSTOMER135 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER136 OFF;
ALTER TABLE CUSTOMER136 DROP CONSTRAINT
CUSTOMER136CKC;
ALTER TABLE CUSTOMER136 ADD CONSTRAINT
CUSTOMER136CKC CHECK (C_W_ID BETWEEN 216001
AND 217600);
SET INTEGRITY FOR CUSTOMER136 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER137 OFF;
ALTER TABLE CUSTOMER137 DROP CONSTRAINT
CUSTOMER137CKC;
ALTER TABLE CUSTOMER137 ADD CONSTRAINT
CUSTOMER137CKC CHECK (C_W_ID BETWEEN 217601
AND 219200);
SET INTEGRITY FOR CUSTOMER137 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER138 OFF;
ALTER TABLE CUSTOMER138 DROP CONSTRAINT
CUSTOMER138CKC;
ALTER TABLE CUSTOMER138 ADD CONSTRAINT
CUSTOMER138CKC CHECK (C_W_ID BETWEEN 219201
AND 220800);
SET INTEGRITY FOR CUSTOMER138 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER139 OFF;
ALTER TABLE CUSTOMER139 DROP CONSTRAINT
CUSTOMER139CKC;
ALTER TABLE CUSTOMER139 ADD CONSTRAINT
CUSTOMER139CKC CHECK (C_W_ID BETWEEN 220801
AND 222400);
SET INTEGRITY FOR CUSTOMER139 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER140 OFF;
ALTER TABLE CUSTOMER140 DROP CONSTRAINT
CUSTOMER140CKC;
ALTER TABLE CUSTOMER140 ADD CONSTRAINT
CUSTOMER140CKC CHECK (C_W_ID BETWEEN 222401
AND 224000);
SET INTEGRITY FOR CUSTOMER140 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER141 OFF;
ALTER TABLE CUSTOMER141 DROP CONSTRAINT
CUSTOMER141CKC;

```

```

ALTER TABLE CUSTOMER141 ADD CONSTRAINT
CUSTOMER141CKC CHECK (C_W_ID BETWEEN 224001
AND 225600);
SET INTEGRITY FOR CUSTOMER141 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER142 OFF;
ALTER TABLE CUSTOMER142 DROP CONSTRAINT
CUSTOMER142CKC;
ALTER TABLE CUSTOMER142 ADD CONSTRAINT
CUSTOMER142CKC CHECK (C_W_ID BETWEEN 225601
AND 227200);
SET INTEGRITY FOR CUSTOMER142 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER143 OFF;
ALTER TABLE CUSTOMER143 DROP CONSTRAINT
CUSTOMER143CKC;
ALTER TABLE CUSTOMER143 ADD CONSTRAINT
CUSTOMER143CKC CHECK (C_W_ID BETWEEN 227201
AND 228800);
SET INTEGRITY FOR CUSTOMER143 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER144 OFF;
ALTER TABLE CUSTOMER144 DROP CONSTRAINT
CUSTOMER144CKC;
ALTER TABLE CUSTOMER144 ADD CONSTRAINT
CUSTOMER144CKC CHECK (C_W_ID BETWEEN 228801
AND 230400);
SET INTEGRITY FOR CUSTOMER144 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER145 OFF;
ALTER TABLE CUSTOMER145 DROP CONSTRAINT
CUSTOMER145CKC;
ALTER TABLE CUSTOMER145 ADD CONSTRAINT
CUSTOMER145CKC CHECK (C_W_ID BETWEEN 230401
AND 232000);
SET INTEGRITY FOR CUSTOMER145 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER146 OFF;
ALTER TABLE CUSTOMER146 DROP CONSTRAINT
CUSTOMER146CKC;
ALTER TABLE CUSTOMER146 ADD CONSTRAINT
CUSTOMER146CKC CHECK (C_W_ID BETWEEN 232001
AND 233600);
SET INTEGRITY FOR CUSTOMER146 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER147 OFF;
ALTER TABLE CUSTOMER147 DROP CONSTRAINT
CUSTOMER147CKC;

```

```

ALTER TABLE CUSTOMER147 ADD CONSTRAINT
CUSTOMER147CKC CHECK (C_W_ID BETWEEN 233601
AND 235200);
SET INTEGRITY FOR CUSTOMER147 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER148 OFF;
ALTER TABLE CUSTOMER148 DROP CONSTRAINT
CUSTOMER148CKC;
ALTER TABLE CUSTOMER148 ADD CONSTRAINT
CUSTOMER148CKC CHECK (C_W_ID BETWEEN 235201
AND 236800);
SET INTEGRITY FOR CUSTOMER148 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER149 OFF;
ALTER TABLE CUSTOMER149 DROP CONSTRAINT
CUSTOMER149CKC;
ALTER TABLE CUSTOMER149 ADD CONSTRAINT
CUSTOMER149CKC CHECK (C_W_ID BETWEEN 236801
AND 238400);
SET INTEGRITY FOR CUSTOMER149 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER150 OFF;
ALTER TABLE CUSTOMER150 DROP CONSTRAINT
CUSTOMER150CKC;
ALTER TABLE CUSTOMER150 ADD CONSTRAINT
CUSTOMER150CKC CHECK (C_W_ID BETWEEN 238401
AND 240000);
SET INTEGRITY FOR CUSTOMER150 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER151 OFF;
ALTER TABLE CUSTOMER151 DROP CONSTRAINT
CUSTOMER151CKC;
ALTER TABLE CUSTOMER151 ADD CONSTRAINT
CUSTOMER151CKC CHECK (C_W_ID BETWEEN 240001
AND 241600);
SET INTEGRITY FOR CUSTOMER151 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER152 OFF;
ALTER TABLE CUSTOMER152 DROP CONSTRAINT
CUSTOMER152CKC;
ALTER TABLE CUSTOMER152 ADD CONSTRAINT
CUSTOMER152CKC CHECK (C_W_ID BETWEEN 241601
AND 243200);
SET INTEGRITY FOR CUSTOMER152 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER153 OFF;
ALTER TABLE CUSTOMER153 DROP CONSTRAINT
CUSTOMER153CKC;

```

```

ALTER TABLE CUSTOMER153 ADD CONSTRAINT
CUSTOMER153CKC CHECK (C_W_ID BETWEEN 243201
AND 244800);
SET INTEGRITY FOR CUSTOMER153 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER154 OFF;
ALTER TABLE CUSTOMER154 DROP CONSTRAINT
CUSTOMER154CKC;
ALTER TABLE CUSTOMER154 ADD CONSTRAINT
CUSTOMER154CKC CHECK (C_W_ID BETWEEN 244801
AND 246400);
SET INTEGRITY FOR CUSTOMER154 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER155 OFF;
ALTER TABLE CUSTOMER155 DROP CONSTRAINT
CUSTOMER155CKC;
ALTER TABLE CUSTOMER155 ADD CONSTRAINT
CUSTOMER155CKC CHECK (C_W_ID BETWEEN 246401
AND 248000);
SET INTEGRITY FOR CUSTOMER155 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER156 OFF;
ALTER TABLE CUSTOMER156 DROP CONSTRAINT
CUSTOMER156CKC;
ALTER TABLE CUSTOMER156 ADD CONSTRAINT
CUSTOMER156CKC CHECK (C_W_ID BETWEEN 248001
AND 249600);
SET INTEGRITY FOR CUSTOMER156 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER157 OFF;
ALTER TABLE CUSTOMER157 DROP CONSTRAINT
CUSTOMER157CKC;
ALTER TABLE CUSTOMER157 ADD CONSTRAINT
CUSTOMER157CKC CHECK (C_W_ID BETWEEN 249601
AND 251200);
SET INTEGRITY FOR CUSTOMER157 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER158 OFF;
ALTER TABLE CUSTOMER158 DROP CONSTRAINT
CUSTOMER158CKC;
ALTER TABLE CUSTOMER158 ADD CONSTRAINT
CUSTOMER158CKC CHECK (C_W_ID BETWEEN 251201
AND 252800);
SET INTEGRITY FOR CUSTOMER158 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER159 OFF;
ALTER TABLE CUSTOMER159 DROP CONSTRAINT
CUSTOMER159CKC;

```

```

ALTER TABLE CUSTOMER159 ADD CONSTRAINT
CUSTOMER159CKC CHECK (C_W_ID BETWEEN 252801
AND 254400);
SET INTEGRITY FOR CUSTOMER159 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER160 OFF;
ALTER TABLE CUSTOMER160 DROP CONSTRAINT
CUSTOMER160CKC;
ALTER TABLE CUSTOMER160 ADD CONSTRAINT
CUSTOMER160CKC CHECK (C_W_ID >= 254401);
SET INTEGRITY FOR CUSTOMER160 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

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 32000);
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 32001 AND
64000);
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 64001 AND
96000);
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 96001 AND
128000);
SET INTEGRITY FOR DISTRICT4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT5 OFF;
ALTER TABLE DISTRICT5 DROP CONSTRAINT
DISTRICT5CKC;

```

```

ALTER TABLE DISTRICT5 ADD CONSTRAINT
DISTRICT5CKC CHECK (D_W_ID BETWEEN 128001 AND
160000);
SET INTEGRITY FOR DISTRICT5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT6 OFF;
ALTER TABLE DISTRICT6 DROP CONSTRAINT
DISTRICT6CKC;
ALTER TABLE DISTRICT6 ADD CONSTRAINT
DISTRICT6CKC CHECK (D_W_ID BETWEEN 160001 AND
192000);
SET INTEGRITY FOR DISTRICT6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT7 OFF;
ALTER TABLE DISTRICT7 DROP CONSTRAINT
DISTRICT7CKC;
ALTER TABLE DISTRICT7 ADD CONSTRAINT
DISTRICT7CKC CHECK (D_W_ID BETWEEN 192001 AND
224000);
SET INTEGRITY FOR DISTRICT7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT8 OFF;
ALTER TABLE DISTRICT8 DROP CONSTRAINT
DISTRICT8CKC;
ALTER TABLE DISTRICT8 ADD CONSTRAINT
DISTRICT8CKC CHECK (D_W_ID >= 224001);
SET INTEGRITY FOR DISTRICT8 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRCONST HISTORY.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY1 OFF;
ALTER TABLE HISTORY1 DROP CONSTRAINT
HISTORY1CKC;
ALTER TABLE HISTORY1 ADD CONSTRAINT
HISTORY1CKC CHECK (H_W_ID BETWEEN 1 AND 16000);
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 16001 AND
32000);
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 32001 AND
48000);
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 48001 AND
64000);
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 64001 AND
80000);
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 80001 AND
96000);
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 96001 AND
112000);
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 112001 AND
128000);
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 128001 AND
144000);
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 144001 AND
160000);
SET INTEGRITY FOR HISTORY10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY11 OFF;
ALTER TABLE HISTORY11 DROP CONSTRAINT
HISTORY11CKC;
ALTER TABLE HISTORY11 ADD CONSTRAINT
HISTORY11CKC CHECK (H_W_ID BETWEEN 160001 AND
176000);
SET INTEGRITY FOR HISTORY11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY12 OFF;
ALTER TABLE HISTORY12 DROP CONSTRAINT
HISTORY12CKC;
ALTER TABLE HISTORY12 ADD CONSTRAINT
HISTORY12CKC CHECK (H_W_ID BETWEEN 176001 AND
192000);
SET INTEGRITY FOR HISTORY12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY13 OFF;
ALTER TABLE HISTORY13 DROP CONSTRAINT
HISTORY13CKC;
ALTER TABLE HISTORY13 ADD CONSTRAINT
HISTORY13CKC CHECK (H_W_ID BETWEEN 192001 AND
208000);
SET INTEGRITY FOR HISTORY13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY14 OFF;
ALTER TABLE HISTORY14 DROP CONSTRAINT
HISTORY14CKC;
ALTER TABLE HISTORY14 ADD CONSTRAINT
HISTORY14CKC CHECK (H_W_ID BETWEEN 208001 AND
224000);
SET INTEGRITY FOR HISTORY14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY15 OFF;
ALTER TABLE HISTORY15 DROP CONSTRAINT
HISTORY15CKC;

```

```

ALTER TABLE HISTORY15 ADD CONSTRAINT
HISTORY15CKC CHECK (H_W_ID BETWEEN 224001 AND
240000);
SET INTEGRITY FOR HISTORY15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY16 OFF;
ALTER TABLE HISTORY16 DROP CONSTRAINT
HISTORY16CKC;
ALTER TABLE HISTORY16 ADD CONSTRAINT
HISTORY16CKC CHECK (H_W_ID >= 240001);
SET INTEGRITY FOR HISTORY16 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRCONST NEW ORDER.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
32000) AND (NO_O_ID <= 3694));
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 32001
AND 64000) AND (NO_O_ID <= 3694));
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 64001
AND 96000) AND (NO_O_ID <= 3694));
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 96001
AND 128000) AND (NO_O_ID <= 3694));
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 128001
AND 160000) AND (NO_O_ID <= 3694));
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 160001
AND 192000) AND (NO_O_ID <= 3694));
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 192001
AND 224000) AND (NO_O_ID <= 3694));
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 >= 224001) AND
(NO_O_ID <= 3694));
SET INTEGRITY FOR NEW_ORDERA8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB1 OFF;
ALTER TABLE NEW_ORDERB1 DROP CONSTRAINT
NEW_ORDERB1CKC;
ALTER TABLE NEW_ORDERB1 ADD CONSTRAINT
NEW_ORDERB1CKC CHECK ((NO_W_ID BETWEEN 1 AND
32000) AND (NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB2 OFF;
ALTER TABLE NEW_ORDERB2 DROP CONSTRAINT
NEW_ORDERB2CKC;
ALTER TABLE NEW_ORDERB2 ADD CONSTRAINT
NEW_ORDERB2CKC CHECK ((NO_W_ID BETWEEN 32001
AND 64000) AND (NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB3 OFF;
ALTER TABLE NEW_ORDERB3 DROP CONSTRAINT
NEW_ORDERB3CKC;

```

```

ALTER TABLE NEW_ORDERB3 ADD CONSTRAINT
NEW_ORDERB3CKC CHECK ((NO_W_ID BETWEEN 64001
AND 96000) AND (NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB4 OFF;
ALTER TABLE NEW_ORDERB4 DROP CONSTRAINT
NEW_ORDERB4CKC;
ALTER TABLE NEW_ORDERB4 ADD CONSTRAINT
NEW_ORDERB4CKC CHECK ((NO_W_ID BETWEEN 96001
AND 128000) AND (NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB5 OFF;
ALTER TABLE NEW_ORDERB5 DROP CONSTRAINT
NEW_ORDERB5CKC;
ALTER TABLE NEW_ORDERB5 ADD CONSTRAINT
NEW_ORDERB5CKC CHECK ((NO_W_ID BETWEEN 128001
AND 160000) AND (NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB6 OFF;
ALTER TABLE NEW_ORDERB6 DROP CONSTRAINT
NEW_ORDERB6CKC;
ALTER TABLE NEW_ORDERB6 ADD CONSTRAINT
NEW_ORDERB6CKC CHECK ((NO_W_ID BETWEEN 160001
AND 192000) AND (NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB7 OFF;
ALTER TABLE NEW_ORDERB7 DROP CONSTRAINT
NEW_ORDERB7CKC;
ALTER TABLE NEW_ORDERB7 ADD CONSTRAINT
NEW_ORDERB7CKC CHECK ((NO_W_ID BETWEEN 192001
AND 224000) AND (NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDERB8 OFF;
ALTER TABLE NEW_ORDERB8 DROP CONSTRAINT
NEW_ORDERB8CKC;
ALTER TABLE NEW_ORDERB8 ADD CONSTRAINT
NEW_ORDERB8CKC CHECK ((NO_W_ID >= 224001) AND
(NO_O_ID >= 3695));
SET INTEGRITY FOR NEW_ORDERB8 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

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 1600);
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 1601 AND 3200);
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 3201 AND 4800);
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 4801 AND 6400);
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 6401 AND 8000);
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 8001 AND 9600);
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 9601 AND 11200);
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 11201 AND 12800);
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 12801 AND 14400);
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 14401 AND
16000);
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 16001 AND
17600);
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 17601 AND
19200);
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 19201 AND
20800);
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 20801 AND
22400);
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 22401 AND
24000);
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 24001 AND
25600);
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 25601 AND
27200);
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 27201 AND
28800);
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 28801 AND
30400);
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 30401 AND
32000);
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 32001 AND
33600);
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 33601 AND
35200);
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 35201 AND
36800);
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 36801 AND
38400);
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 38401 AND
40000);
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 40001 AND
41600);
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 41601 AND
43200);
SET INTEGRITY FOR ORDERS27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS28 OFF;
ALTER TABLE ORDERS28 DROP CONSTRAINT
ORDERS28CKC;
ALTER TABLE ORDERS28 ADD CONSTRAINT
ORDERS28CKC CHECK (O_W_ID BETWEEN 43201 AND
44800);
SET INTEGRITY FOR ORDERS28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS29 OFF;
ALTER TABLE ORDERS29 DROP CONSTRAINT
ORDERS29CKC;
ALTER TABLE ORDERS29 ADD CONSTRAINT
ORDERS29CKC CHECK (O_W_ID BETWEEN 44801 AND
46400);
SET INTEGRITY FOR ORDERS29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS30 OFF;
ALTER TABLE ORDERS30 DROP CONSTRAINT
ORDERS30CKC;
ALTER TABLE ORDERS30 ADD CONSTRAINT
ORDERS30CKC CHECK (O_W_ID BETWEEN 46401 AND
48000);
SET INTEGRITY FOR ORDERS30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS31 OFF;
ALTER TABLE ORDERS31 DROP CONSTRAINT
ORDERS31CKC;
ALTER TABLE ORDERS31 ADD CONSTRAINT
ORDERS31CKC CHECK (O_W_ID BETWEEN 48001 AND
49600);
SET INTEGRITY FOR ORDERS31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS32 OFF;
ALTER TABLE ORDERS32 DROP CONSTRAINT
ORDERS32CKC;

```

```

ALTER TABLE ORDERS32 ADD CONSTRAINT
ORDERS32CKC CHECK (O_W_ID BETWEEN 49601 AND
51200);
SET INTEGRITY FOR ORDERS32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS33 OFF;
ALTER TABLE ORDERS33 DROP CONSTRAINT
ORDERS33CKC;
ALTER TABLE ORDERS33 ADD CONSTRAINT
ORDERS33CKC CHECK (O_W_ID BETWEEN 51201 AND
52800);
SET INTEGRITY FOR ORDERS33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS34 OFF;
ALTER TABLE ORDERS34 DROP CONSTRAINT
ORDERS34CKC;
ALTER TABLE ORDERS34 ADD CONSTRAINT
ORDERS34CKC CHECK (O_W_ID BETWEEN 52801 AND
54400);
SET INTEGRITY FOR ORDERS34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS35 OFF;
ALTER TABLE ORDERS35 DROP CONSTRAINT
ORDERS35CKC;
ALTER TABLE ORDERS35 ADD CONSTRAINT
ORDERS35CKC CHECK (O_W_ID BETWEEN 54401 AND
56000);
SET INTEGRITY FOR ORDERS35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS36 OFF;
ALTER TABLE ORDERS36 DROP CONSTRAINT
ORDERS36CKC;
ALTER TABLE ORDERS36 ADD CONSTRAINT
ORDERS36CKC CHECK (O_W_ID BETWEEN 56001 AND
57600);
SET INTEGRITY FOR ORDERS36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS37 OFF;
ALTER TABLE ORDERS37 DROP CONSTRAINT
ORDERS37CKC;
ALTER TABLE ORDERS37 ADD CONSTRAINT
ORDERS37CKC CHECK (O_W_ID BETWEEN 57601 AND
59200);
SET INTEGRITY FOR ORDERS37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS38 OFF;
ALTER TABLE ORDERS38 DROP CONSTRAINT
ORDERS38CKC;

```

```

ALTER TABLE ORDERS38 ADD CONSTRAINT
ORDERS38CKC CHECK (O_W_ID BETWEEN 59201 AND
60800);
SET INTEGRITY FOR ORDERS38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS39 OFF;
ALTER TABLE ORDERS39 DROP CONSTRAINT
ORDERS39CKC;
ALTER TABLE ORDERS39 ADD CONSTRAINT
ORDERS39CKC CHECK (O_W_ID BETWEEN 60801 AND
62400);
SET INTEGRITY FOR ORDERS39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS40 OFF;
ALTER TABLE ORDERS40 DROP CONSTRAINT
ORDERS40CKC;
ALTER TABLE ORDERS40 ADD CONSTRAINT
ORDERS40CKC CHECK (O_W_ID BETWEEN 62401 AND
64000);
SET INTEGRITY FOR ORDERS40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS41 OFF;
ALTER TABLE ORDERS41 DROP CONSTRAINT
ORDERS41CKC;
ALTER TABLE ORDERS41 ADD CONSTRAINT
ORDERS41CKC CHECK (O_W_ID BETWEEN 64001 AND
65600);
SET INTEGRITY FOR ORDERS41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS42 OFF;
ALTER TABLE ORDERS42 DROP CONSTRAINT
ORDERS42CKC;
ALTER TABLE ORDERS42 ADD CONSTRAINT
ORDERS42CKC CHECK (O_W_ID BETWEEN 65601 AND
67200);
SET INTEGRITY FOR ORDERS42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS43 OFF;
ALTER TABLE ORDERS43 DROP CONSTRAINT
ORDERS43CKC;
ALTER TABLE ORDERS43 ADD CONSTRAINT
ORDERS43CKC CHECK (O_W_ID BETWEEN 67201 AND
68800);
SET INTEGRITY FOR ORDERS43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS44 OFF;
ALTER TABLE ORDERS44 DROP CONSTRAINT
ORDERS44CKC;

```

```

ALTER TABLE ORDERS44 ADD CONSTRAINT
ORDERS44CKC CHECK (O_W_ID BETWEEN 68801 AND
70400);
SET INTEGRITY FOR ORDERS44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS45 OFF;
ALTER TABLE ORDERS45 DROP CONSTRAINT
ORDERS45CKC;
ALTER TABLE ORDERS45 ADD CONSTRAINT
ORDERS45CKC CHECK (O_W_ID BETWEEN 70401 AND
72000);
SET INTEGRITY FOR ORDERS45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS46 OFF;
ALTER TABLE ORDERS46 DROP CONSTRAINT
ORDERS46CKC;
ALTER TABLE ORDERS46 ADD CONSTRAINT
ORDERS46CKC CHECK (O_W_ID BETWEEN 72001 AND
73600);
SET INTEGRITY FOR ORDERS46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS47 OFF;
ALTER TABLE ORDERS47 DROP CONSTRAINT
ORDERS47CKC;
ALTER TABLE ORDERS47 ADD CONSTRAINT
ORDERS47CKC CHECK (O_W_ID BETWEEN 73601 AND
75200);
SET INTEGRITY FOR ORDERS47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS48 OFF;
ALTER TABLE ORDERS48 DROP CONSTRAINT
ORDERS48CKC;
ALTER TABLE ORDERS48 ADD CONSTRAINT
ORDERS48CKC CHECK (O_W_ID BETWEEN 75201 AND
76800);
SET INTEGRITY FOR ORDERS48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS49 OFF;
ALTER TABLE ORDERS49 DROP CONSTRAINT
ORDERS49CKC;
ALTER TABLE ORDERS49 ADD CONSTRAINT
ORDERS49CKC CHECK (O_W_ID BETWEEN 76801 AND
78400);
SET INTEGRITY FOR ORDERS49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS50 OFF;
ALTER TABLE ORDERS50 DROP CONSTRAINT
ORDERS50CKC;

```

```

ALTER TABLE ORDERS50 ADD CONSTRAINT
ORDERS50CKC CHECK (O_W_ID BETWEEN 78401 AND
80000);
SET INTEGRITY FOR ORDERS50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS51 OFF;
ALTER TABLE ORDERS51 DROP CONSTRAINT
ORDERS51CKC;
ALTER TABLE ORDERS51 ADD CONSTRAINT
ORDERS51CKC CHECK (O_W_ID BETWEEN 80001 AND
81600);
SET INTEGRITY FOR ORDERS51 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS52 OFF;
ALTER TABLE ORDERS52 DROP CONSTRAINT
ORDERS52CKC;
ALTER TABLE ORDERS52 ADD CONSTRAINT
ORDERS52CKC CHECK (O_W_ID BETWEEN 81601 AND
83200);
SET INTEGRITY FOR ORDERS52 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS53 OFF;
ALTER TABLE ORDERS53 DROP CONSTRAINT
ORDERS53CKC;
ALTER TABLE ORDERS53 ADD CONSTRAINT
ORDERS53CKC CHECK (O_W_ID BETWEEN 83201 AND
84800);
SET INTEGRITY FOR ORDERS53 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS54 OFF;
ALTER TABLE ORDERS54 DROP CONSTRAINT
ORDERS54CKC;
ALTER TABLE ORDERS54 ADD CONSTRAINT
ORDERS54CKC CHECK (O_W_ID BETWEEN 84801 AND
86400);
SET INTEGRITY FOR ORDERS54 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS55 OFF;
ALTER TABLE ORDERS55 DROP CONSTRAINT
ORDERS55CKC;
ALTER TABLE ORDERS55 ADD CONSTRAINT
ORDERS55CKC CHECK (O_W_ID BETWEEN 86401 AND
88000);
SET INTEGRITY FOR ORDERS55 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS56 OFF;
ALTER TABLE ORDERS56 DROP CONSTRAINT
ORDERS56CKC;

```

```

ALTER TABLE ORDERS56 ADD CONSTRAINT
ORDERS56CKC CHECK (O_W_ID BETWEEN 88001 AND
89600);
SET INTEGRITY FOR ORDERS56 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS57 OFF;
ALTER TABLE ORDERS57 DROP CONSTRAINT
ORDERS57CKC;
ALTER TABLE ORDERS57 ADD CONSTRAINT
ORDERS57CKC CHECK (O_W_ID BETWEEN 89601 AND
91200);
SET INTEGRITY FOR ORDERS57 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS58 OFF;
ALTER TABLE ORDERS58 DROP CONSTRAINT
ORDERS58CKC;
ALTER TABLE ORDERS58 ADD CONSTRAINT
ORDERS58CKC CHECK (O_W_ID BETWEEN 91201 AND
92800);
SET INTEGRITY FOR ORDERS58 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS59 OFF;
ALTER TABLE ORDERS59 DROP CONSTRAINT
ORDERS59CKC;
ALTER TABLE ORDERS59 ADD CONSTRAINT
ORDERS59CKC CHECK (O_W_ID BETWEEN 92801 AND
94400);
SET INTEGRITY FOR ORDERS59 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS60 OFF;
ALTER TABLE ORDERS60 DROP CONSTRAINT
ORDERS60CKC;
ALTER TABLE ORDERS60 ADD CONSTRAINT
ORDERS60CKC CHECK (O_W_ID BETWEEN 94401 AND
96000);
SET INTEGRITY FOR ORDERS60 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS61 OFF;
ALTER TABLE ORDERS61 DROP CONSTRAINT
ORDERS61CKC;
ALTER TABLE ORDERS61 ADD CONSTRAINT
ORDERS61CKC CHECK (O_W_ID BETWEEN 96001 AND
97600);
SET INTEGRITY FOR ORDERS61 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS62 OFF;
ALTER TABLE ORDERS62 DROP CONSTRAINT
ORDERS62CKC;

```

```

ALTER TABLE ORDERS62 ADD CONSTRAINT
ORDERS62CKC CHECK (O_W_ID BETWEEN 97601 AND
99200);
SET INTEGRITY FOR ORDERS62 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS63 OFF;
ALTER TABLE ORDERS63 DROP CONSTRAINT
ORDERS63CKC;
ALTER TABLE ORDERS63 ADD CONSTRAINT
ORDERS63CKC CHECK (O_W_ID BETWEEN 99201 AND
100800);
SET INTEGRITY FOR ORDERS63 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS64 OFF;
ALTER TABLE ORDERS64 DROP CONSTRAINT
ORDERS64CKC;
ALTER TABLE ORDERS64 ADD CONSTRAINT
ORDERS64CKC CHECK (O_W_ID BETWEEN 100801 AND
102400);
SET INTEGRITY FOR ORDERS64 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS65 OFF;
ALTER TABLE ORDERS65 DROP CONSTRAINT
ORDERS65CKC;
ALTER TABLE ORDERS65 ADD CONSTRAINT
ORDERS65CKC CHECK (O_W_ID BETWEEN 102401 AND
104000);
SET INTEGRITY FOR ORDERS65 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS66 OFF;
ALTER TABLE ORDERS66 DROP CONSTRAINT
ORDERS66CKC;
ALTER TABLE ORDERS66 ADD CONSTRAINT
ORDERS66CKC CHECK (O_W_ID BETWEEN 104001 AND
105600);
SET INTEGRITY FOR ORDERS66 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS67 OFF;
ALTER TABLE ORDERS67 DROP CONSTRAINT
ORDERS67CKC;
ALTER TABLE ORDERS67 ADD CONSTRAINT
ORDERS67CKC CHECK (O_W_ID BETWEEN 105601 AND
107200);
SET INTEGRITY FOR ORDERS67 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS68 OFF;
ALTER TABLE ORDERS68 DROP CONSTRAINT
ORDERS68CKC;

```

```

ALTER TABLE ORDERS68 ADD CONSTRAINT
ORDERS68CKC CHECK (O_W_ID BETWEEN 107201 AND
108800);
SET INTEGRITY FOR ORDERS68 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS69 OFF;
ALTER TABLE ORDERS69 DROP CONSTRAINT
ORDERS69CKC;
ALTER TABLE ORDERS69 ADD CONSTRAINT
ORDERS69CKC CHECK (O_W_ID BETWEEN 108801 AND
110400);
SET INTEGRITY FOR ORDERS69 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS70 OFF;
ALTER TABLE ORDERS70 DROP CONSTRAINT
ORDERS70CKC;
ALTER TABLE ORDERS70 ADD CONSTRAINT
ORDERS70CKC CHECK (O_W_ID BETWEEN 110401 AND
112000);
SET INTEGRITY FOR ORDERS70 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS71 OFF;
ALTER TABLE ORDERS71 DROP CONSTRAINT
ORDERS71CKC;
ALTER TABLE ORDERS71 ADD CONSTRAINT
ORDERS71CKC CHECK (O_W_ID BETWEEN 112001 AND
113600);
SET INTEGRITY FOR ORDERS71 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS72 OFF;
ALTER TABLE ORDERS72 DROP CONSTRAINT
ORDERS72CKC;
ALTER TABLE ORDERS72 ADD CONSTRAINT
ORDERS72CKC CHECK (O_W_ID BETWEEN 113601 AND
115200);
SET INTEGRITY FOR ORDERS72 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS73 OFF;
ALTER TABLE ORDERS73 DROP CONSTRAINT
ORDERS73CKC;
ALTER TABLE ORDERS73 ADD CONSTRAINT
ORDERS73CKC CHECK (O_W_ID BETWEEN 115201 AND
116800);
SET INTEGRITY FOR ORDERS73 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS74 OFF;
ALTER TABLE ORDERS74 DROP CONSTRAINT
ORDERS74CKC;

```

```

ALTER TABLE ORDERS74 ADD CONSTRAINT
ORDERS74CKC CHECK (O_W_ID BETWEEN 116801 AND
118400);
SET INTEGRITY FOR ORDERS74 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS75 OFF;
ALTER TABLE ORDERS75 DROP CONSTRAINT
ORDERS75CKC;
ALTER TABLE ORDERS75 ADD CONSTRAINT
ORDERS75CKC CHECK (O_W_ID BETWEEN 118401 AND
120000);
SET INTEGRITY FOR ORDERS75 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS76 OFF;
ALTER TABLE ORDERS76 DROP CONSTRAINT
ORDERS76CKC;
ALTER TABLE ORDERS76 ADD CONSTRAINT
ORDERS76CKC CHECK (O_W_ID BETWEEN 120001 AND
121600);
SET INTEGRITY FOR ORDERS76 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS77 OFF;
ALTER TABLE ORDERS77 DROP CONSTRAINT
ORDERS77CKC;
ALTER TABLE ORDERS77 ADD CONSTRAINT
ORDERS77CKC CHECK (O_W_ID BETWEEN 121601 AND
123200);
SET INTEGRITY FOR ORDERS77 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS78 OFF;
ALTER TABLE ORDERS78 DROP CONSTRAINT
ORDERS78CKC;
ALTER TABLE ORDERS78 ADD CONSTRAINT
ORDERS78CKC CHECK (O_W_ID BETWEEN 123201 AND
124800);
SET INTEGRITY FOR ORDERS78 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS79 OFF;
ALTER TABLE ORDERS79 DROP CONSTRAINT
ORDERS79CKC;
ALTER TABLE ORDERS79 ADD CONSTRAINT
ORDERS79CKC CHECK (O_W_ID BETWEEN 124801 AND
126400);
SET INTEGRITY FOR ORDERS79 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS80 OFF;
ALTER TABLE ORDERS80 DROP CONSTRAINT
ORDERS80CKC;

```

```

ALTER TABLE ORDERS80 ADD CONSTRAINT
ORDERS80CKC CHECK (O_W_ID BETWEEN 126401 AND
128000);
SET INTEGRITY FOR ORDERS80 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS81 OFF;
ALTER TABLE ORDERS81 DROP CONSTRAINT
ORDERS81CKC;
ALTER TABLE ORDERS81 ADD CONSTRAINT
ORDERS81CKC CHECK (O_W_ID BETWEEN 128001 AND
129600);
SET INTEGRITY FOR ORDERS81 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS82 OFF;
ALTER TABLE ORDERS82 DROP CONSTRAINT
ORDERS82CKC;
ALTER TABLE ORDERS82 ADD CONSTRAINT
ORDERS82CKC CHECK (O_W_ID BETWEEN 129601 AND
131200);
SET INTEGRITY FOR ORDERS82 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS83 OFF;
ALTER TABLE ORDERS83 DROP CONSTRAINT
ORDERS83CKC;
ALTER TABLE ORDERS83 ADD CONSTRAINT
ORDERS83CKC CHECK (O_W_ID BETWEEN 131201 AND
132800);
SET INTEGRITY FOR ORDERS83 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS84 OFF;
ALTER TABLE ORDERS84 DROP CONSTRAINT
ORDERS84CKC;
ALTER TABLE ORDERS84 ADD CONSTRAINT
ORDERS84CKC CHECK (O_W_ID BETWEEN 132801 AND
134400);
SET INTEGRITY FOR ORDERS84 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS85 OFF;
ALTER TABLE ORDERS85 DROP CONSTRAINT
ORDERS85CKC;
ALTER TABLE ORDERS85 ADD CONSTRAINT
ORDERS85CKC CHECK (O_W_ID BETWEEN 134401 AND
136000);
SET INTEGRITY FOR ORDERS85 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS86 OFF;
ALTER TABLE ORDERS86 DROP CONSTRAINT
ORDERS86CKC;

```

```

ALTER TABLE ORDERS86 ADD CONSTRAINT
ORDERS86CKC CHECK (O_W_ID BETWEEN 136001 AND
137600);
SET INTEGRITY FOR ORDERS86 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS87 OFF;
ALTER TABLE ORDERS87 DROP CONSTRAINT
ORDERS87CKC;
ALTER TABLE ORDERS87 ADD CONSTRAINT
ORDERS87CKC CHECK (O_W_ID BETWEEN 137601 AND
139200);
SET INTEGRITY FOR ORDERS87 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS88 OFF;
ALTER TABLE ORDERS88 DROP CONSTRAINT
ORDERS88CKC;
ALTER TABLE ORDERS88 ADD CONSTRAINT
ORDERS88CKC CHECK (O_W_ID BETWEEN 139201 AND
140800);
SET INTEGRITY FOR ORDERS88 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS89 OFF;
ALTER TABLE ORDERS89 DROP CONSTRAINT
ORDERS89CKC;
ALTER TABLE ORDERS89 ADD CONSTRAINT
ORDERS89CKC CHECK (O_W_ID BETWEEN 140801 AND
142400);
SET INTEGRITY FOR ORDERS89 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS90 OFF;
ALTER TABLE ORDERS90 DROP CONSTRAINT
ORDERS90CKC;
ALTER TABLE ORDERS90 ADD CONSTRAINT
ORDERS90CKC CHECK (O_W_ID BETWEEN 142401 AND
144000);
SET INTEGRITY FOR ORDERS90 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS91 OFF;
ALTER TABLE ORDERS91 DROP CONSTRAINT
ORDERS91CKC;
ALTER TABLE ORDERS91 ADD CONSTRAINT
ORDERS91CKC CHECK (O_W_ID BETWEEN 144001 AND
145600);
SET INTEGRITY FOR ORDERS91 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS92 OFF;
ALTER TABLE ORDERS92 DROP CONSTRAINT
ORDERS92CKC;

```

```

ALTER TABLE ORDERS92 ADD CONSTRAINT
ORDERS92CKC CHECK (O_W_ID BETWEEN 145601 AND
147200);
SET INTEGRITY FOR ORDERS92 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS93 OFF;
ALTER TABLE ORDERS93 DROP CONSTRAINT
ORDERS93CKC;
ALTER TABLE ORDERS93 ADD CONSTRAINT
ORDERS93CKC CHECK (O_W_ID BETWEEN 147201 AND
148800);
SET INTEGRITY FOR ORDERS93 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS94 OFF;
ALTER TABLE ORDERS94 DROP CONSTRAINT
ORDERS94CKC;
ALTER TABLE ORDERS94 ADD CONSTRAINT
ORDERS94CKC CHECK (O_W_ID BETWEEN 148801 AND
150400);
SET INTEGRITY FOR ORDERS94 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS95 OFF;
ALTER TABLE ORDERS95 DROP CONSTRAINT
ORDERS95CKC;
ALTER TABLE ORDERS95 ADD CONSTRAINT
ORDERS95CKC CHECK (O_W_ID BETWEEN 150401 AND
152000);
SET INTEGRITY FOR ORDERS95 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS96 OFF;
ALTER TABLE ORDERS96 DROP CONSTRAINT
ORDERS96CKC;
ALTER TABLE ORDERS96 ADD CONSTRAINT
ORDERS96CKC CHECK (O_W_ID BETWEEN 152001 AND
153600);
SET INTEGRITY FOR ORDERS96 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS97 OFF;
ALTER TABLE ORDERS97 DROP CONSTRAINT
ORDERS97CKC;
ALTER TABLE ORDERS97 ADD CONSTRAINT
ORDERS97CKC CHECK (O_W_ID BETWEEN 153601 AND
155200);
SET INTEGRITY FOR ORDERS97 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS98 OFF;
ALTER TABLE ORDERS98 DROP CONSTRAINT
ORDERS98CKC;

```

```

ALTER TABLE ORDERS98 ADD CONSTRAINT
ORDERS98CKC CHECK (O_W_ID BETWEEN 155201 AND
156800);
SET INTEGRITY FOR ORDERS98 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS99 OFF;
ALTER TABLE ORDERS99 DROP CONSTRAINT
ORDERS99CKC;
ALTER TABLE ORDERS99 ADD CONSTRAINT
ORDERS99CKC CHECK (O_W_ID BETWEEN 156801 AND
158400);
SET INTEGRITY FOR ORDERS99 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS100 OFF;
ALTER TABLE ORDERS100 DROP CONSTRAINT
ORDERS100CKC;
ALTER TABLE ORDERS100 ADD CONSTRAINT
ORDERS100CKC CHECK (O_W_ID BETWEEN 158401 AND
160000);
SET INTEGRITY FOR ORDERS100 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS101 OFF;
ALTER TABLE ORDERS101 DROP CONSTRAINT
ORDERS101CKC;
ALTER TABLE ORDERS101 ADD CONSTRAINT
ORDERS101CKC CHECK (O_W_ID BETWEEN 160001 AND
161600);
SET INTEGRITY FOR ORDERS101 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS102 OFF;
ALTER TABLE ORDERS102 DROP CONSTRAINT
ORDERS102CKC;
ALTER TABLE ORDERS102 ADD CONSTRAINT
ORDERS102CKC CHECK (O_W_ID BETWEEN 161601 AND
163200);
SET INTEGRITY FOR ORDERS102 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS103 OFF;
ALTER TABLE ORDERS103 DROP CONSTRAINT
ORDERS103CKC;
ALTER TABLE ORDERS103 ADD CONSTRAINT
ORDERS103CKC CHECK (O_W_ID BETWEEN 163201 AND
164800);
SET INTEGRITY FOR ORDERS103 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS104 OFF;
ALTER TABLE ORDERS104 DROP CONSTRAINT
ORDERS104CKC;

```

```

ALTER TABLE ORDERS104 ADD CONSTRAINT
ORDERS104CKC CHECK (O_W_ID BETWEEN 164801 AND
166400);
SET INTEGRITY FOR ORDERS104 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS105 OFF;
ALTER TABLE ORDERS105 DROP CONSTRAINT
ORDERS105CKC;
ALTER TABLE ORDERS105 ADD CONSTRAINT
ORDERS105CKC CHECK (O_W_ID BETWEEN 166401 AND
168000);
SET INTEGRITY FOR ORDERS105 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS106 OFF;
ALTER TABLE ORDERS106 DROP CONSTRAINT
ORDERS106CKC;
ALTER TABLE ORDERS106 ADD CONSTRAINT
ORDERS106CKC CHECK (O_W_ID BETWEEN 168001 AND
169600);
SET INTEGRITY FOR ORDERS106 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS107 OFF;
ALTER TABLE ORDERS107 DROP CONSTRAINT
ORDERS107CKC;
ALTER TABLE ORDERS107 ADD CONSTRAINT
ORDERS107CKC CHECK (O_W_ID BETWEEN 169601 AND
171200);
SET INTEGRITY FOR ORDERS107 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS108 OFF;
ALTER TABLE ORDERS108 DROP CONSTRAINT
ORDERS108CKC;
ALTER TABLE ORDERS108 ADD CONSTRAINT
ORDERS108CKC CHECK (O_W_ID BETWEEN 171201 AND
172800);
SET INTEGRITY FOR ORDERS108 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS109 OFF;
ALTER TABLE ORDERS109 DROP CONSTRAINT
ORDERS109CKC;
ALTER TABLE ORDERS109 ADD CONSTRAINT
ORDERS109CKC CHECK (O_W_ID BETWEEN 172801 AND
174400);
SET INTEGRITY FOR ORDERS109 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS110 OFF;
ALTER TABLE ORDERS110 DROP CONSTRAINT
ORDERS110CKC;

```

```

ALTER TABLE ORDERS110 ADD CONSTRAINT
ORDERS110CKC CHECK (O_W_ID BETWEEN 174401 AND
176000);
SET INTEGRITY FOR ORDERS110 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS111 OFF;
ALTER TABLE ORDERS111 DROP CONSTRAINT
ORDERS111CKC;
ALTER TABLE ORDERS111 ADD CONSTRAINT
ORDERS111CKC CHECK (O_W_ID BETWEEN 176001 AND
177600);
SET INTEGRITY FOR ORDERS111 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS112 OFF;
ALTER TABLE ORDERS112 DROP CONSTRAINT
ORDERS112CKC;
ALTER TABLE ORDERS112 ADD CONSTRAINT
ORDERS112CKC CHECK (O_W_ID BETWEEN 177601 AND
179200);
SET INTEGRITY FOR ORDERS112 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS113 OFF;
ALTER TABLE ORDERS113 DROP CONSTRAINT
ORDERS113CKC;
ALTER TABLE ORDERS113 ADD CONSTRAINT
ORDERS113CKC CHECK (O_W_ID BETWEEN 179201 AND
180800);
SET INTEGRITY FOR ORDERS113 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS114 OFF;
ALTER TABLE ORDERS114 DROP CONSTRAINT
ORDERS114CKC;
ALTER TABLE ORDERS114 ADD CONSTRAINT
ORDERS114CKC CHECK (O_W_ID BETWEEN 180801 AND
182400);
SET INTEGRITY FOR ORDERS114 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS115 OFF;
ALTER TABLE ORDERS115 DROP CONSTRAINT
ORDERS115CKC;
ALTER TABLE ORDERS115 ADD CONSTRAINT
ORDERS115CKC CHECK (O_W_ID BETWEEN 182401 AND
184000);
SET INTEGRITY FOR ORDERS115 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS116 OFF;
ALTER TABLE ORDERS116 DROP CONSTRAINT
ORDERS116CKC;

```



```

ALTER TABLE ORDERS116 ADD CONSTRAINT
ORDERS116CKC CHECK (O_W_ID BETWEEN 184001 AND
185600);
SET INTEGRITY FOR ORDERS116 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS117 OFF;
ALTER TABLE ORDERS117 DROP CONSTRAINT
ORDERS117CKC;
ALTER TABLE ORDERS117 ADD CONSTRAINT
ORDERS117CKC CHECK (O_W_ID BETWEEN 185601 AND
187200);
SET INTEGRITY FOR ORDERS117 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS118 OFF;
ALTER TABLE ORDERS118 DROP CONSTRAINT
ORDERS118CKC;
ALTER TABLE ORDERS118 ADD CONSTRAINT
ORDERS118CKC CHECK (O_W_ID BETWEEN 187201 AND
188800);
SET INTEGRITY FOR ORDERS118 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS119 OFF;
ALTER TABLE ORDERS119 DROP CONSTRAINT
ORDERS119CKC;
ALTER TABLE ORDERS119 ADD CONSTRAINT
ORDERS119CKC CHECK (O_W_ID BETWEEN 188801 AND
190400);
SET INTEGRITY FOR ORDERS119 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS120 OFF;
ALTER TABLE ORDERS120 DROP CONSTRAINT
ORDERS120CKC;
ALTER TABLE ORDERS120 ADD CONSTRAINT
ORDERS120CKC CHECK (O_W_ID BETWEEN 190401 AND
192000);
SET INTEGRITY FOR ORDERS120 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS121 OFF;
ALTER TABLE ORDERS121 DROP CONSTRAINT
ORDERS121CKC;
ALTER TABLE ORDERS121 ADD CONSTRAINT
ORDERS121CKC CHECK (O_W_ID BETWEEN 192001 AND
193600);
SET INTEGRITY FOR ORDERS121 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS122 OFF;
ALTER TABLE ORDERS122 DROP CONSTRAINT
ORDERS122CKC;

```

```

ALTER TABLE ORDERS122 ADD CONSTRAINT
ORDERS122CKC CHECK (O_W_ID BETWEEN 193601 AND
195200);
SET INTEGRITY FOR ORDERS122 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS123 OFF;
ALTER TABLE ORDERS123 DROP CONSTRAINT
ORDERS123CKC;
ALTER TABLE ORDERS123 ADD CONSTRAINT
ORDERS123CKC CHECK (O_W_ID BETWEEN 195201 AND
196800);
SET INTEGRITY FOR ORDERS123 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS124 OFF;
ALTER TABLE ORDERS124 DROP CONSTRAINT
ORDERS124CKC;
ALTER TABLE ORDERS124 ADD CONSTRAINT
ORDERS124CKC CHECK (O_W_ID BETWEEN 196801 AND
198400);
SET INTEGRITY FOR ORDERS124 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS125 OFF;
ALTER TABLE ORDERS125 DROP CONSTRAINT
ORDERS125CKC;
ALTER TABLE ORDERS125 ADD CONSTRAINT
ORDERS125CKC CHECK (O_W_ID BETWEEN 198401 AND
200000);
SET INTEGRITY FOR ORDERS125 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS126 OFF;
ALTER TABLE ORDERS126 DROP CONSTRAINT
ORDERS126CKC;
ALTER TABLE ORDERS126 ADD CONSTRAINT
ORDERS126CKC CHECK (O_W_ID BETWEEN 200001 AND
201600);
SET INTEGRITY FOR ORDERS126 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS127 OFF;
ALTER TABLE ORDERS127 DROP CONSTRAINT
ORDERS127CKC;
ALTER TABLE ORDERS127 ADD CONSTRAINT
ORDERS127CKC CHECK (O_W_ID BETWEEN 201601 AND
203200);
SET INTEGRITY FOR ORDERS127 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS128 OFF;
ALTER TABLE ORDERS128 DROP CONSTRAINT
ORDERS128CKC;

```

```

ALTER TABLE ORDERS128 ADD CONSTRAINT
ORDERS128CKC CHECK (O_W_ID BETWEEN 203201 AND
204800);
SET INTEGRITY FOR ORDERS128 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS129 OFF;
ALTER TABLE ORDERS129 DROP CONSTRAINT
ORDERS129CKC;
ALTER TABLE ORDERS129 ADD CONSTRAINT
ORDERS129CKC CHECK (O_W_ID BETWEEN 204801 AND
206400);
SET INTEGRITY FOR ORDERS129 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS130 OFF;
ALTER TABLE ORDERS130 DROP CONSTRAINT
ORDERS130CKC;
ALTER TABLE ORDERS130 ADD CONSTRAINT
ORDERS130CKC CHECK (O_W_ID BETWEEN 206401 AND
208000);
SET INTEGRITY FOR ORDERS130 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS131 OFF;
ALTER TABLE ORDERS131 DROP CONSTRAINT
ORDERS131CKC;
ALTER TABLE ORDERS131 ADD CONSTRAINT
ORDERS131CKC CHECK (O_W_ID BETWEEN 208001 AND
209600);
SET INTEGRITY FOR ORDERS131 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS132 OFF;
ALTER TABLE ORDERS132 DROP CONSTRAINT
ORDERS132CKC;
ALTER TABLE ORDERS132 ADD CONSTRAINT
ORDERS132CKC CHECK (O_W_ID BETWEEN 209601 AND
211200);
SET INTEGRITY FOR ORDERS132 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS133 OFF;
ALTER TABLE ORDERS133 DROP CONSTRAINT
ORDERS133CKC;
ALTER TABLE ORDERS133 ADD CONSTRAINT
ORDERS133CKC CHECK (O_W_ID BETWEEN 211201 AND
212800);
SET INTEGRITY FOR ORDERS133 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS134 OFF;
ALTER TABLE ORDERS134 DROP CONSTRAINT
ORDERS134CKC;

```

```

ALTER TABLE ORDERS134 ADD CONSTRAINT
ORDERS134CKC CHECK (O_W_ID BETWEEN 212801 AND
214400);
SET INTEGRITY FOR ORDERS134 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS135 OFF;
ALTER TABLE ORDERS135 DROP CONSTRAINT
ORDERS135CKC;
ALTER TABLE ORDERS135 ADD CONSTRAINT
ORDERS135CKC CHECK (O_W_ID BETWEEN 214401 AND
216000);
SET INTEGRITY FOR ORDERS135 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS136 OFF;
ALTER TABLE ORDERS136 DROP CONSTRAINT
ORDERS136CKC;
ALTER TABLE ORDERS136 ADD CONSTRAINT
ORDERS136CKC CHECK (O_W_ID BETWEEN 216001 AND
217600);
SET INTEGRITY FOR ORDERS136 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS137 OFF;
ALTER TABLE ORDERS137 DROP CONSTRAINT
ORDERS137CKC;
ALTER TABLE ORDERS137 ADD CONSTRAINT
ORDERS137CKC CHECK (O_W_ID BETWEEN 217601 AND
219200);
SET INTEGRITY FOR ORDERS137 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS138 OFF;
ALTER TABLE ORDERS138 DROP CONSTRAINT
ORDERS138CKC;
ALTER TABLE ORDERS138 ADD CONSTRAINT
ORDERS138CKC CHECK (O_W_ID BETWEEN 219201 AND
220800);
SET INTEGRITY FOR ORDERS138 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS139 OFF;
ALTER TABLE ORDERS139 DROP CONSTRAINT
ORDERS139CKC;
ALTER TABLE ORDERS139 ADD CONSTRAINT
ORDERS139CKC CHECK (O_W_ID BETWEEN 220801 AND
222400);
SET INTEGRITY FOR ORDERS139 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS140 OFF;
ALTER TABLE ORDERS140 DROP CONSTRAINT
ORDERS140CKC;

```

```

ALTER TABLE ORDERS140 ADD CONSTRAINT
ORDERS140CKC CHECK (O_W_ID BETWEEN 222401 AND
224000);
SET INTEGRITY FOR ORDERS140 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS141 OFF;
ALTER TABLE ORDERS141 DROP CONSTRAINT
ORDERS141CKC;
ALTER TABLE ORDERS141 ADD CONSTRAINT
ORDERS141CKC CHECK (O_W_ID BETWEEN 224001 AND
225600);
SET INTEGRITY FOR ORDERS141 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS142 OFF;
ALTER TABLE ORDERS142 DROP CONSTRAINT
ORDERS142CKC;
ALTER TABLE ORDERS142 ADD CONSTRAINT
ORDERS142CKC CHECK (O_W_ID BETWEEN 225601 AND
227200);
SET INTEGRITY FOR ORDERS142 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS143 OFF;
ALTER TABLE ORDERS143 DROP CONSTRAINT
ORDERS143CKC;
ALTER TABLE ORDERS143 ADD CONSTRAINT
ORDERS143CKC CHECK (O_W_ID BETWEEN 227201 AND
228800);
SET INTEGRITY FOR ORDERS143 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS144 OFF;
ALTER TABLE ORDERS144 DROP CONSTRAINT
ORDERS144CKC;
ALTER TABLE ORDERS144 ADD CONSTRAINT
ORDERS144CKC CHECK (O_W_ID BETWEEN 228801 AND
230400);
SET INTEGRITY FOR ORDERS144 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS145 OFF;
ALTER TABLE ORDERS145 DROP CONSTRAINT
ORDERS145CKC;
ALTER TABLE ORDERS145 ADD CONSTRAINT
ORDERS145CKC CHECK (O_W_ID BETWEEN 230401 AND
232000);
SET INTEGRITY FOR ORDERS145 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS146 OFF;
ALTER TABLE ORDERS146 DROP CONSTRAINT
ORDERS146CKC;

```

```

ALTER TABLE ORDERS146 ADD CONSTRAINT
ORDERS146CKC CHECK (O_W_ID BETWEEN 232001 AND
233600);
SET INTEGRITY FOR ORDERS146 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS147 OFF;
ALTER TABLE ORDERS147 DROP CONSTRAINT
ORDERS147CKC;
ALTER TABLE ORDERS147 ADD CONSTRAINT
ORDERS147CKC CHECK (O_W_ID BETWEEN 233601 AND
235200);
SET INTEGRITY FOR ORDERS147 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS148 OFF;
ALTER TABLE ORDERS148 DROP CONSTRAINT
ORDERS148CKC;
ALTER TABLE ORDERS148 ADD CONSTRAINT
ORDERS148CKC CHECK (O_W_ID BETWEEN 235201 AND
236800);
SET INTEGRITY FOR ORDERS148 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS149 OFF;
ALTER TABLE ORDERS149 DROP CONSTRAINT
ORDERS149CKC;
ALTER TABLE ORDERS149 ADD CONSTRAINT
ORDERS149CKC CHECK (O_W_ID BETWEEN 236801 AND
238400);
SET INTEGRITY FOR ORDERS149 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS150 OFF;
ALTER TABLE ORDERS150 DROP CONSTRAINT
ORDERS150CKC;
ALTER TABLE ORDERS150 ADD CONSTRAINT
ORDERS150CKC CHECK (O_W_ID BETWEEN 238401 AND
240000);
SET INTEGRITY FOR ORDERS150 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS151 OFF;
ALTER TABLE ORDERS151 DROP CONSTRAINT
ORDERS151CKC;
ALTER TABLE ORDERS151 ADD CONSTRAINT
ORDERS151CKC CHECK (O_W_ID BETWEEN 240001 AND
241600);
SET INTEGRITY FOR ORDERS151 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS152 OFF;
ALTER TABLE ORDERS152 DROP CONSTRAINT
ORDERS152CKC;

```

```

ALTER TABLE ORDERS152 ADD CONSTRAINT
ORDERS152CKC CHECK (O_W_ID BETWEEN 241601 AND
243200);
SET INTEGRITY FOR ORDERS152 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS153 OFF;
ALTER TABLE ORDERS153 DROP CONSTRAINT
ORDERS153CKC;
ALTER TABLE ORDERS153 ADD CONSTRAINT
ORDERS153CKC CHECK (O_W_ID BETWEEN 243201 AND
244800);
SET INTEGRITY FOR ORDERS153 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS154 OFF;
ALTER TABLE ORDERS154 DROP CONSTRAINT
ORDERS154CKC;
ALTER TABLE ORDERS154 ADD CONSTRAINT
ORDERS154CKC CHECK (O_W_ID BETWEEN 244801 AND
246400);
SET INTEGRITY FOR ORDERS154 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS155 OFF;
ALTER TABLE ORDERS155 DROP CONSTRAINT
ORDERS155CKC;
ALTER TABLE ORDERS155 ADD CONSTRAINT
ORDERS155CKC CHECK (O_W_ID BETWEEN 246401 AND
248000);
SET INTEGRITY FOR ORDERS155 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS156 OFF;
ALTER TABLE ORDERS156 DROP CONSTRAINT
ORDERS156CKC;
ALTER TABLE ORDERS156 ADD CONSTRAINT
ORDERS156CKC CHECK (O_W_ID BETWEEN 248001 AND
249600);
SET INTEGRITY FOR ORDERS156 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS157 OFF;
ALTER TABLE ORDERS157 DROP CONSTRAINT
ORDERS157CKC;
ALTER TABLE ORDERS157 ADD CONSTRAINT
ORDERS157CKC CHECK (O_W_ID BETWEEN 249601 AND
251200);
SET INTEGRITY FOR ORDERS157 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS158 OFF;
ALTER TABLE ORDERS158 DROP CONSTRAINT
ORDERS158CKC;

```

```

ALTER TABLE ORDERS158 ADD CONSTRAINT
ORDERS158CKC CHECK (O_W_ID BETWEEN 251201 AND
252800);
SET INTEGRITY FOR ORDERS158 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS159 OFF;
ALTER TABLE ORDERS159 DROP CONSTRAINT
ORDERS159CKC;
ALTER TABLE ORDERS159 ADD CONSTRAINT
ORDERS159CKC CHECK (O_W_ID BETWEEN 252801 AND
254400);
SET INTEGRITY FOR ORDERS159 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS160 OFF;
ALTER TABLE ORDERS160 DROP CONSTRAINT
ORDERS160CKC;
ALTER TABLE ORDERS160 ADD CONSTRAINT
ORDERS160CKC CHECK (O_W_ID >= 254401);
SET INTEGRITY FOR ORDERS160 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRCONST ORDER LINE.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE1 OFF;
ALTER TABLE ORDER_LINE1 DROP CONSTRAINT
ORDER_LINE1CKC;
ALTER TABLE ORDER_LINE1 ADD CONSTRAINT
ORDER_LINE1CKC CHECK (OL_W_ID BETWEEN 1 AND
1600);
SET INTEGRITY FOR ORDER_LINE1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE2 OFF;
ALTER TABLE ORDER_LINE2 DROP CONSTRAINT
ORDER_LINE2CKC;
ALTER TABLE ORDER_LINE2 ADD CONSTRAINT
ORDER_LINE2CKC CHECK (OL_W_ID BETWEEN 1601 AND
3200);
SET INTEGRITY FOR ORDER_LINE2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE3 OFF;
ALTER TABLE ORDER_LINE3 DROP CONSTRAINT
ORDER_LINE3CKC;
ALTER TABLE ORDER_LINE3 ADD CONSTRAINT
ORDER_LINE3CKC CHECK (OL_W_ID BETWEEN 3201 AND
4800);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE4 OFF;

```

```

ALTER TABLE ORDER_LINE4 DROP CONSTRAINT
ORDER_LINE4CKC;
ALTER TABLE ORDER_LINE4 ADD CONSTRAINT
ORDER_LINE4CKC CHECK (OL_W_ID BETWEEN 4801 AND
6400);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE5 OFF;
ALTER TABLE ORDER_LINE5 DROP CONSTRAINT
ORDER_LINE5CKC;
ALTER TABLE ORDER_LINE5 ADD CONSTRAINT
ORDER_LINE5CKC CHECK (OL_W_ID BETWEEN 6401 AND
8000);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE6 OFF;
ALTER TABLE ORDER_LINE6 DROP CONSTRAINT
ORDER_LINE6CKC;
ALTER TABLE ORDER_LINE6 ADD CONSTRAINT
ORDER_LINE6CKC CHECK (OL_W_ID BETWEEN 8001 AND
9600);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE7 OFF;
ALTER TABLE ORDER_LINE7 DROP CONSTRAINT
ORDER_LINE7CKC;
ALTER TABLE ORDER_LINE7 ADD CONSTRAINT
ORDER_LINE7CKC CHECK (OL_W_ID BETWEEN 9601 AND
11200);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE8 OFF;
ALTER TABLE ORDER_LINE8 DROP CONSTRAINT
ORDER_LINE8CKC;
ALTER TABLE ORDER_LINE8 ADD CONSTRAINT
ORDER_LINE8CKC CHECK (OL_W_ID BETWEEN 11201
AND 12800);
SET INTEGRITY FOR ORDER_LINE8 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE9 OFF;
ALTER TABLE ORDER_LINE9 DROP CONSTRAINT
ORDER_LINE9CKC;
ALTER TABLE ORDER_LINE9 ADD CONSTRAINT
ORDER_LINE9CKC CHECK (OL_W_ID BETWEEN 12801
AND 14400);
SET INTEGRITY FOR ORDER_LINE9 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE10 OFF;
ALTER TABLE ORDER_LINE10 DROP CONSTRAINT
ORDER_LINE10CKC;

```

```

ALTER TABLE ORDER_LINE10 ADD CONSTRAINT
ORDER_LINE10CKC CHECK (OL_W_ID BETWEEN 14401
AND 16000);
SET INTEGRITY FOR ORDER_LINE10 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE11 OFF;
ALTER TABLE ORDER_LINE11 DROP CONSTRAINT
ORDER_LINE11CKC;
ALTER TABLE ORDER_LINE11 ADD CONSTRAINT
ORDER_LINE11CKC CHECK (OL_W_ID BETWEEN 16001
AND 17600);
SET INTEGRITY FOR ORDER_LINE11 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE12 OFF;
ALTER TABLE ORDER_LINE12 DROP CONSTRAINT
ORDER_LINE12CKC;
ALTER TABLE ORDER_LINE12 ADD CONSTRAINT
ORDER_LINE12CKC CHECK (OL_W_ID BETWEEN 17601
AND 19200);
SET INTEGRITY FOR ORDER_LINE12 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE13 OFF;
ALTER TABLE ORDER_LINE13 DROP CONSTRAINT
ORDER_LINE13CKC;
ALTER TABLE ORDER_LINE13 ADD CONSTRAINT
ORDER_LINE13CKC CHECK (OL_W_ID BETWEEN 19201
AND 20800);
SET INTEGRITY FOR ORDER_LINE13 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE14 OFF;
ALTER TABLE ORDER_LINE14 DROP CONSTRAINT
ORDER_LINE14CKC;
ALTER TABLE ORDER_LINE14 ADD CONSTRAINT
ORDER_LINE14CKC CHECK (OL_W_ID BETWEEN 20801
AND 22400);
SET INTEGRITY FOR ORDER_LINE14 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE15 OFF;
ALTER TABLE ORDER_LINE15 DROP CONSTRAINT
ORDER_LINE15CKC;
ALTER TABLE ORDER_LINE15 ADD CONSTRAINT
ORDER_LINE15CKC CHECK (OL_W_ID BETWEEN 22401
AND 24000);
SET INTEGRITY FOR ORDER_LINE15 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE16 OFF;
ALTER TABLE ORDER_LINE16 DROP CONSTRAINT
ORDER_LINE16CKC;

```

```

ALTER TABLE ORDER_LINE16 ADD CONSTRAINT
ORDER_LINE16CKC CHECK (OL_W_ID BETWEEN 24001
AND 25600);
SET INTEGRITY FOR ORDER_LINE16 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE17 OFF;
ALTER TABLE ORDER_LINE17 DROP CONSTRAINT
ORDER_LINE17CKC;
ALTER TABLE ORDER_LINE17 ADD CONSTRAINT
ORDER_LINE17CKC CHECK (OL_W_ID BETWEEN 25601
AND 27200);
SET INTEGRITY FOR ORDER_LINE17 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE18 OFF;
ALTER TABLE ORDER_LINE18 DROP CONSTRAINT
ORDER_LINE18CKC;
ALTER TABLE ORDER_LINE18 ADD CONSTRAINT
ORDER_LINE18CKC CHECK (OL_W_ID BETWEEN 27201
AND 28800);
SET INTEGRITY FOR ORDER_LINE18 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE19 OFF;
ALTER TABLE ORDER_LINE19 DROP CONSTRAINT
ORDER_LINE19CKC;
ALTER TABLE ORDER_LINE19 ADD CONSTRAINT
ORDER_LINE19CKC CHECK (OL_W_ID BETWEEN 28801
AND 30400);
SET INTEGRITY FOR ORDER_LINE19 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE20 OFF;
ALTER TABLE ORDER_LINE20 DROP CONSTRAINT
ORDER_LINE20CKC;
ALTER TABLE ORDER_LINE20 ADD CONSTRAINT
ORDER_LINE20CKC CHECK (OL_W_ID BETWEEN 30401
AND 32000);
SET INTEGRITY FOR ORDER_LINE20 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE21 OFF;
ALTER TABLE ORDER_LINE21 DROP CONSTRAINT
ORDER_LINE21CKC;
ALTER TABLE ORDER_LINE21 ADD CONSTRAINT
ORDER_LINE21CKC CHECK (OL_W_ID BETWEEN 32001
AND 33600);
SET INTEGRITY FOR ORDER_LINE21 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE22 OFF;
ALTER TABLE ORDER_LINE22 DROP CONSTRAINT
ORDER_LINE22CKC;

```

```

ALTER TABLE ORDER_LINE22 ADD CONSTRAINT
ORDER_LINE22CKC CHECK (OL_W_ID BETWEEN 33601
AND 35200);
SET INTEGRITY FOR ORDER_LINE22 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE23 OFF;
ALTER TABLE ORDER_LINE23 DROP CONSTRAINT
ORDER_LINE23CKC;
ALTER TABLE ORDER_LINE23 ADD CONSTRAINT
ORDER_LINE23CKC CHECK (OL_W_ID BETWEEN 35201
AND 36800);
SET INTEGRITY FOR ORDER_LINE23 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE24 OFF;
ALTER TABLE ORDER_LINE24 DROP CONSTRAINT
ORDER_LINE24CKC;
ALTER TABLE ORDER_LINE24 ADD CONSTRAINT
ORDER_LINE24CKC CHECK (OL_W_ID BETWEEN 36801
AND 38400);
SET INTEGRITY FOR ORDER_LINE24 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE25 OFF;
ALTER TABLE ORDER_LINE25 DROP CONSTRAINT
ORDER_LINE25CKC;
ALTER TABLE ORDER_LINE25 ADD CONSTRAINT
ORDER_LINE25CKC CHECK (OL_W_ID BETWEEN 38401
AND 40000);
SET INTEGRITY FOR ORDER_LINE25 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE26 OFF;
ALTER TABLE ORDER_LINE26 DROP CONSTRAINT
ORDER_LINE26CKC;
ALTER TABLE ORDER_LINE26 ADD CONSTRAINT
ORDER_LINE26CKC CHECK (OL_W_ID BETWEEN 40001
AND 41600);
SET INTEGRITY FOR ORDER_LINE26 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE27 OFF;
ALTER TABLE ORDER_LINE27 DROP CONSTRAINT
ORDER_LINE27CKC;
ALTER TABLE ORDER_LINE27 ADD CONSTRAINT
ORDER_LINE27CKC CHECK (OL_W_ID BETWEEN 41601
AND 43200);
SET INTEGRITY FOR ORDER_LINE27 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE28 OFF;
ALTER TABLE ORDER_LINE28 DROP CONSTRAINT
ORDER_LINE28CKC;

```

```

ALTER TABLE ORDER_LINE28 ADD CONSTRAINT
ORDER_LINE28CKC CHECK (OL_W_ID BETWEEN 43201
AND 44800);
SET INTEGRITY FOR ORDER_LINE28 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE29 OFF;
ALTER TABLE ORDER_LINE29 DROP CONSTRAINT
ORDER_LINE29CKC;
ALTER TABLE ORDER_LINE29 ADD CONSTRAINT
ORDER_LINE29CKC CHECK (OL_W_ID BETWEEN 44801
AND 46400);
SET INTEGRITY FOR ORDER_LINE29 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE30 OFF;
ALTER TABLE ORDER_LINE30 DROP CONSTRAINT
ORDER_LINE30CKC;
ALTER TABLE ORDER_LINE30 ADD CONSTRAINT
ORDER_LINE30CKC CHECK (OL_W_ID BETWEEN 46401
AND 48000);
SET INTEGRITY FOR ORDER_LINE30 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE31 OFF;
ALTER TABLE ORDER_LINE31 DROP CONSTRAINT
ORDER_LINE31CKC;
ALTER TABLE ORDER_LINE31 ADD CONSTRAINT
ORDER_LINE31CKC CHECK (OL_W_ID BETWEEN 48001
AND 49600);
SET INTEGRITY FOR ORDER_LINE31 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE32 OFF;
ALTER TABLE ORDER_LINE32 DROP CONSTRAINT
ORDER_LINE32CKC;
ALTER TABLE ORDER_LINE32 ADD CONSTRAINT
ORDER_LINE32CKC CHECK (OL_W_ID BETWEEN 49601
AND 51200);
SET INTEGRITY FOR ORDER_LINE32 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE33 OFF;
ALTER TABLE ORDER_LINE33 DROP CONSTRAINT
ORDER_LINE33CKC;
ALTER TABLE ORDER_LINE33 ADD CONSTRAINT
ORDER_LINE33CKC CHECK (OL_W_ID BETWEEN 51201
AND 52800);
SET INTEGRITY FOR ORDER_LINE33 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE34 OFF;
ALTER TABLE ORDER_LINE34 DROP CONSTRAINT
ORDER_LINE34CKC;

```

```

ALTER TABLE ORDER_LINE34 ADD CONSTRAINT
ORDER_LINE34CKC CHECK (OL_W_ID BETWEEN 52801
AND 54400);
SET INTEGRITY FOR ORDER_LINE34 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE35 OFF;
ALTER TABLE ORDER_LINE35 DROP CONSTRAINT
ORDER_LINE35CKC;
ALTER TABLE ORDER_LINE35 ADD CONSTRAINT
ORDER_LINE35CKC CHECK (OL_W_ID BETWEEN 54401
AND 56000);
SET INTEGRITY FOR ORDER_LINE35 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE36 OFF;
ALTER TABLE ORDER_LINE36 DROP CONSTRAINT
ORDER_LINE36CKC;
ALTER TABLE ORDER_LINE36 ADD CONSTRAINT
ORDER_LINE36CKC CHECK (OL_W_ID BETWEEN 56001
AND 57600);
SET INTEGRITY FOR ORDER_LINE36 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE37 OFF;
ALTER TABLE ORDER_LINE37 DROP CONSTRAINT
ORDER_LINE37CKC;
ALTER TABLE ORDER_LINE37 ADD CONSTRAINT
ORDER_LINE37CKC CHECK (OL_W_ID BETWEEN 57601
AND 59200);
SET INTEGRITY FOR ORDER_LINE37 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE38 OFF;
ALTER TABLE ORDER_LINE38 DROP CONSTRAINT
ORDER_LINE38CKC;
ALTER TABLE ORDER_LINE38 ADD CONSTRAINT
ORDER_LINE38CKC CHECK (OL_W_ID BETWEEN 59201
AND 60800);
SET INTEGRITY FOR ORDER_LINE38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE39 OFF;
ALTER TABLE ORDER_LINE39 DROP CONSTRAINT
ORDER_LINE39CKC;
ALTER TABLE ORDER_LINE39 ADD CONSTRAINT
ORDER_LINE39CKC CHECK (OL_W_ID BETWEEN 60801
AND 62400);
SET INTEGRITY FOR ORDER_LINE39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE40 OFF;
ALTER TABLE ORDER_LINE40 DROP CONSTRAINT
ORDER_LINE40CKC;

```

```

ALTER TABLE ORDER_LINE40 ADD CONSTRAINT
ORDER_LINE40CKC CHECK (OL_W_ID BETWEEN 62401
AND 64000);
SET INTEGRITY FOR ORDER_LINE40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE41 OFF;
ALTER TABLE ORDER_LINE41 DROP CONSTRAINT
ORDER_LINE41CKC;
ALTER TABLE ORDER_LINE41 ADD CONSTRAINT
ORDER_LINE41CKC CHECK (OL_W_ID BETWEEN 64001
AND 65600);
SET INTEGRITY FOR ORDER_LINE41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE42 OFF;
ALTER TABLE ORDER_LINE42 DROP CONSTRAINT
ORDER_LINE42CKC;
ALTER TABLE ORDER_LINE42 ADD CONSTRAINT
ORDER_LINE42CKC CHECK (OL_W_ID BETWEEN 65601
AND 67200);
SET INTEGRITY FOR ORDER_LINE42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE43 OFF;
ALTER TABLE ORDER_LINE43 DROP CONSTRAINT
ORDER_LINE43CKC;
ALTER TABLE ORDER_LINE43 ADD CONSTRAINT
ORDER_LINE43CKC CHECK (OL_W_ID BETWEEN 67201
AND 68800);
SET INTEGRITY FOR ORDER_LINE43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE44 OFF;
ALTER TABLE ORDER_LINE44 DROP CONSTRAINT
ORDER_LINE44CKC;
ALTER TABLE ORDER_LINE44 ADD CONSTRAINT
ORDER_LINE44CKC CHECK (OL_W_ID BETWEEN 68801
AND 70400);
SET INTEGRITY FOR ORDER_LINE44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE45 OFF;
ALTER TABLE ORDER_LINE45 DROP CONSTRAINT
ORDER_LINE45CKC;
ALTER TABLE ORDER_LINE45 ADD CONSTRAINT
ORDER_LINE45CKC CHECK (OL_W_ID BETWEEN 70401
AND 72000);
SET INTEGRITY FOR ORDER_LINE45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE46 OFF;
ALTER TABLE ORDER_LINE46 DROP CONSTRAINT
ORDER_LINE46CKC;

```

```

ALTER TABLE ORDER_LINE46 ADD CONSTRAINT
ORDER_LINE46CKC CHECK (OL_W_ID BETWEEN 72001
AND 73600);
SET INTEGRITY FOR ORDER_LINE46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE47 OFF;
ALTER TABLE ORDER_LINE47 DROP CONSTRAINT
ORDER_LINE47CKC;
ALTER TABLE ORDER_LINE47 ADD CONSTRAINT
ORDER_LINE47CKC CHECK (OL_W_ID BETWEEN 73601
AND 75200);
SET INTEGRITY FOR ORDER_LINE47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE48 OFF;
ALTER TABLE ORDER_LINE48 DROP CONSTRAINT
ORDER_LINE48CKC;
ALTER TABLE ORDER_LINE48 ADD CONSTRAINT
ORDER_LINE48CKC CHECK (OL_W_ID BETWEEN 75201
AND 76800);
SET INTEGRITY FOR ORDER_LINE48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE49 OFF;
ALTER TABLE ORDER_LINE49 DROP CONSTRAINT
ORDER_LINE49CKC;
ALTER TABLE ORDER_LINE49 ADD CONSTRAINT
ORDER_LINE49CKC CHECK (OL_W_ID BETWEEN 76801
AND 78400);
SET INTEGRITY FOR ORDER_LINE49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE50 OFF;
ALTER TABLE ORDER_LINE50 DROP CONSTRAINT
ORDER_LINE50CKC;
ALTER TABLE ORDER_LINE50 ADD CONSTRAINT
ORDER_LINE50CKC CHECK (OL_W_ID BETWEEN 78401
AND 80000);
SET INTEGRITY FOR ORDER_LINE50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE51 OFF;
ALTER TABLE ORDER_LINE51 DROP CONSTRAINT
ORDER_LINE51CKC;
ALTER TABLE ORDER_LINE51 ADD CONSTRAINT
ORDER_LINE51CKC CHECK (OL_W_ID BETWEEN 80001
AND 81600);
SET INTEGRITY FOR ORDER_LINE51 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE52 OFF;
ALTER TABLE ORDER_LINE52 DROP CONSTRAINT
ORDER_LINE52CKC;

```

```

ALTER TABLE ORDER_LINE52 ADD CONSTRAINT
ORDER_LINE52CKC CHECK (OL_W_ID BETWEEN 81601
AND 83200);
SET INTEGRITY FOR ORDER_LINE52 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE53 OFF;
ALTER TABLE ORDER_LINE53 DROP CONSTRAINT
ORDER_LINE53CKC;
ALTER TABLE ORDER_LINE53 ADD CONSTRAINT
ORDER_LINE53CKC CHECK (OL_W_ID BETWEEN 83201
AND 84800);
SET INTEGRITY FOR ORDER_LINE53 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE54 OFF;
ALTER TABLE ORDER_LINE54 DROP CONSTRAINT
ORDER_LINE54CKC;
ALTER TABLE ORDER_LINE54 ADD CONSTRAINT
ORDER_LINE54CKC CHECK (OL_W_ID BETWEEN 84801
AND 86400);
SET INTEGRITY FOR ORDER_LINE54 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE55 OFF;
ALTER TABLE ORDER_LINE55 DROP CONSTRAINT
ORDER_LINE55CKC;
ALTER TABLE ORDER_LINE55 ADD CONSTRAINT
ORDER_LINE55CKC CHECK (OL_W_ID BETWEEN 86401
AND 88000);
SET INTEGRITY FOR ORDER_LINE55 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE56 OFF;
ALTER TABLE ORDER_LINE56 DROP CONSTRAINT
ORDER_LINE56CKC;
ALTER TABLE ORDER_LINE56 ADD CONSTRAINT
ORDER_LINE56CKC CHECK (OL_W_ID BETWEEN 88001
AND 89600);
SET INTEGRITY FOR ORDER_LINE56 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE57 OFF;
ALTER TABLE ORDER_LINE57 DROP CONSTRAINT
ORDER_LINE57CKC;
ALTER TABLE ORDER_LINE57 ADD CONSTRAINT
ORDER_LINE57CKC CHECK (OL_W_ID BETWEEN 89601
AND 91200);
SET INTEGRITY FOR ORDER_LINE57 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE58 OFF;
ALTER TABLE ORDER_LINE58 DROP CONSTRAINT
ORDER_LINE58CKC;

```

```

ALTER TABLE ORDER_LINE58 ADD CONSTRAINT
ORDER_LINE58CKC CHECK (OL_W_ID BETWEEN 91201
AND 92800);
SET INTEGRITY FOR ORDER_LINE58 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE59 OFF;
ALTER TABLE ORDER_LINE59 DROP CONSTRAINT
ORDER_LINE59CKC;
ALTER TABLE ORDER_LINE59 ADD CONSTRAINT
ORDER_LINE59CKC CHECK (OL_W_ID BETWEEN 92801
AND 94400);
SET INTEGRITY FOR ORDER_LINE59 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE60 OFF;
ALTER TABLE ORDER_LINE60 DROP CONSTRAINT
ORDER_LINE60CKC;
ALTER TABLE ORDER_LINE60 ADD CONSTRAINT
ORDER_LINE60CKC CHECK (OL_W_ID BETWEEN 94401
AND 96000);
SET INTEGRITY FOR ORDER_LINE60 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE61 OFF;
ALTER TABLE ORDER_LINE61 DROP CONSTRAINT
ORDER_LINE61CKC;
ALTER TABLE ORDER_LINE61 ADD CONSTRAINT
ORDER_LINE61CKC CHECK (OL_W_ID BETWEEN 96001
AND 97600);
SET INTEGRITY FOR ORDER_LINE61 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE62 OFF;
ALTER TABLE ORDER_LINE62 DROP CONSTRAINT
ORDER_LINE62CKC;
ALTER TABLE ORDER_LINE62 ADD CONSTRAINT
ORDER_LINE62CKC CHECK (OL_W_ID BETWEEN 97601
AND 99200);
SET INTEGRITY FOR ORDER_LINE62 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE63 OFF;
ALTER TABLE ORDER_LINE63 DROP CONSTRAINT
ORDER_LINE63CKC;
ALTER TABLE ORDER_LINE63 ADD CONSTRAINT
ORDER_LINE63CKC CHECK (OL_W_ID BETWEEN 99201
AND 100800);
SET INTEGRITY FOR ORDER_LINE63 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE64 OFF;
ALTER TABLE ORDER_LINE64 DROP CONSTRAINT
ORDER_LINE64CKC;

```

```

ALTER TABLE ORDER_LINE64 ADD CONSTRAINT
ORDER_LINE64CKC CHECK (OL_W_ID BETWEEN 100801
AND 102400);
SET INTEGRITY FOR ORDER_LINE64 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE65 OFF;
ALTER TABLE ORDER_LINE65 DROP CONSTRAINT
ORDER_LINE65CKC;
ALTER TABLE ORDER_LINE65 ADD CONSTRAINT
ORDER_LINE65CKC CHECK (OL_W_ID BETWEEN 102401
AND 104000);
SET INTEGRITY FOR ORDER_LINE65 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE66 OFF;
ALTER TABLE ORDER_LINE66 DROP CONSTRAINT
ORDER_LINE66CKC;
ALTER TABLE ORDER_LINE66 ADD CONSTRAINT
ORDER_LINE66CKC CHECK (OL_W_ID BETWEEN 104001
AND 105600);
SET INTEGRITY FOR ORDER_LINE66 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE67 OFF;
ALTER TABLE ORDER_LINE67 DROP CONSTRAINT
ORDER_LINE67CKC;
ALTER TABLE ORDER_LINE67 ADD CONSTRAINT
ORDER_LINE67CKC CHECK (OL_W_ID BETWEEN 105601
AND 107200);
SET INTEGRITY FOR ORDER_LINE67 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE68 OFF;
ALTER TABLE ORDER_LINE68 DROP CONSTRAINT
ORDER_LINE68CKC;
ALTER TABLE ORDER_LINE68 ADD CONSTRAINT
ORDER_LINE68CKC CHECK (OL_W_ID BETWEEN 107201
AND 108800);
SET INTEGRITY FOR ORDER_LINE68 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE69 OFF;
ALTER TABLE ORDER_LINE69 DROP CONSTRAINT
ORDER_LINE69CKC;
ALTER TABLE ORDER_LINE69 ADD CONSTRAINT
ORDER_LINE69CKC CHECK (OL_W_ID BETWEEN 108801
AND 110400);
SET INTEGRITY FOR ORDER_LINE69 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE70 OFF;
ALTER TABLE ORDER_LINE70 DROP CONSTRAINT
ORDER_LINE70CKC;

```

```

ALTER TABLE ORDER_LINE70 ADD CONSTRAINT
ORDER_LINE70CKC CHECK (OL_W_ID BETWEEN 110401
AND 112000);
SET INTEGRITY FOR ORDER_LINE70 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE71 OFF;
ALTER TABLE ORDER_LINE71 DROP CONSTRAINT
ORDER_LINE71CKC;
ALTER TABLE ORDER_LINE71 ADD CONSTRAINT
ORDER_LINE71CKC CHECK (OL_W_ID BETWEEN 112001
AND 113600);
SET INTEGRITY FOR ORDER_LINE71 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE72 OFF;
ALTER TABLE ORDER_LINE72 DROP CONSTRAINT
ORDER_LINE72CKC;
ALTER TABLE ORDER_LINE72 ADD CONSTRAINT
ORDER_LINE72CKC CHECK (OL_W_ID BETWEEN 113601
AND 115200);
SET INTEGRITY FOR ORDER_LINE72 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE73 OFF;
ALTER TABLE ORDER_LINE73 DROP CONSTRAINT
ORDER_LINE73CKC;
ALTER TABLE ORDER_LINE73 ADD CONSTRAINT
ORDER_LINE73CKC CHECK (OL_W_ID BETWEEN 115201
AND 116800);
SET INTEGRITY FOR ORDER_LINE73 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE74 OFF;
ALTER TABLE ORDER_LINE74 DROP CONSTRAINT
ORDER_LINE74CKC;
ALTER TABLE ORDER_LINE74 ADD CONSTRAINT
ORDER_LINE74CKC CHECK (OL_W_ID BETWEEN 116801
AND 118400);
SET INTEGRITY FOR ORDER_LINE74 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE75 OFF;
ALTER TABLE ORDER_LINE75 DROP CONSTRAINT
ORDER_LINE75CKC;
ALTER TABLE ORDER_LINE75 ADD CONSTRAINT
ORDER_LINE75CKC CHECK (OL_W_ID BETWEEN 118401
AND 120000);
SET INTEGRITY FOR ORDER_LINE75 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE76 OFF;
ALTER TABLE ORDER_LINE76 DROP CONSTRAINT
ORDER_LINE76CKC;

```

```

ALTER TABLE ORDER_LINE76 ADD CONSTRAINT
ORDER_LINE76CKC CHECK (OL_W_ID BETWEEN 120001
AND 121600);
SET INTEGRITY FOR ORDER_LINE76 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE77 OFF;
ALTER TABLE ORDER_LINE77 DROP CONSTRAINT
ORDER_LINE77CKC;
ALTER TABLE ORDER_LINE77 ADD CONSTRAINT
ORDER_LINE77CKC CHECK (OL_W_ID BETWEEN 121601
AND 123200);
SET INTEGRITY FOR ORDER_LINE77 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE78 OFF;
ALTER TABLE ORDER_LINE78 DROP CONSTRAINT
ORDER_LINE78CKC;
ALTER TABLE ORDER_LINE78 ADD CONSTRAINT
ORDER_LINE78CKC CHECK (OL_W_ID BETWEEN 123201
AND 124800);
SET INTEGRITY FOR ORDER_LINE78 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE79 OFF;
ALTER TABLE ORDER_LINE79 DROP CONSTRAINT
ORDER_LINE79CKC;
ALTER TABLE ORDER_LINE79 ADD CONSTRAINT
ORDER_LINE79CKC CHECK (OL_W_ID BETWEEN 124801
AND 126400);
SET INTEGRITY FOR ORDER_LINE79 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE80 OFF;
ALTER TABLE ORDER_LINE80 DROP CONSTRAINT
ORDER_LINE80CKC;
ALTER TABLE ORDER_LINE80 ADD CONSTRAINT
ORDER_LINE80CKC CHECK (OL_W_ID BETWEEN 126401
AND 128000);
SET INTEGRITY FOR ORDER_LINE80 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE81 OFF;
ALTER TABLE ORDER_LINE81 DROP CONSTRAINT
ORDER_LINE81CKC;
ALTER TABLE ORDER_LINE81 ADD CONSTRAINT
ORDER_LINE81CKC CHECK (OL_W_ID BETWEEN 128001
AND 129600);
SET INTEGRITY FOR ORDER_LINE81 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE82 OFF;
ALTER TABLE ORDER_LINE82 DROP CONSTRAINT
ORDER_LINE82CKC;

```

```

ALTER TABLE ORDER_LINE82 ADD CONSTRAINT
ORDER_LINE82CKC CHECK (OL_W_ID BETWEEN 129601
AND 131200);
SET INTEGRITY FOR ORDER_LINE82 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE83 OFF;
ALTER TABLE ORDER_LINE83 DROP CONSTRAINT
ORDER_LINE83CKC;
ALTER TABLE ORDER_LINE83 ADD CONSTRAINT
ORDER_LINE83CKC CHECK (OL_W_ID BETWEEN 131201
AND 132800);
SET INTEGRITY FOR ORDER_LINE83 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE84 OFF;
ALTER TABLE ORDER_LINE84 DROP CONSTRAINT
ORDER_LINE84CKC;
ALTER TABLE ORDER_LINE84 ADD CONSTRAINT
ORDER_LINE84CKC CHECK (OL_W_ID BETWEEN 132801
AND 134400);
SET INTEGRITY FOR ORDER_LINE84 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE85 OFF;
ALTER TABLE ORDER_LINE85 DROP CONSTRAINT
ORDER_LINE85CKC;
ALTER TABLE ORDER_LINE85 ADD CONSTRAINT
ORDER_LINE85CKC CHECK (OL_W_ID BETWEEN 134401
AND 136000);
SET INTEGRITY FOR ORDER_LINE85 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE86 OFF;
ALTER TABLE ORDER_LINE86 DROP CONSTRAINT
ORDER_LINE86CKC;
ALTER TABLE ORDER_LINE86 ADD CONSTRAINT
ORDER_LINE86CKC CHECK (OL_W_ID BETWEEN 136001
AND 137600);
SET INTEGRITY FOR ORDER_LINE86 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE87 OFF;
ALTER TABLE ORDER_LINE87 DROP CONSTRAINT
ORDER_LINE87CKC;
ALTER TABLE ORDER_LINE87 ADD CONSTRAINT
ORDER_LINE87CKC CHECK (OL_W_ID BETWEEN 137601
AND 139200);
SET INTEGRITY FOR ORDER_LINE87 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE88 OFF;
ALTER TABLE ORDER_LINE88 DROP CONSTRAINT
ORDER_LINE88CKC;

```

```

ALTER TABLE ORDER_LINE88 ADD CONSTRAINT
ORDER_LINE88CKC CHECK (OL_W_ID BETWEEN 139201
AND 140800);
SET INTEGRITY FOR ORDER_LINE88 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE89 OFF;
ALTER TABLE ORDER_LINE89 DROP CONSTRAINT
ORDER_LINE89CKC;
ALTER TABLE ORDER_LINE89 ADD CONSTRAINT
ORDER_LINE89CKC CHECK (OL_W_ID BETWEEN 140801
AND 142400);
SET INTEGRITY FOR ORDER_LINE89 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE90 OFF;
ALTER TABLE ORDER_LINE90 DROP CONSTRAINT
ORDER_LINE90CKC;
ALTER TABLE ORDER_LINE90 ADD CONSTRAINT
ORDER_LINE90CKC CHECK (OL_W_ID BETWEEN 142401
AND 144000);
SET INTEGRITY FOR ORDER_LINE90 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE91 OFF;
ALTER TABLE ORDER_LINE91 DROP CONSTRAINT
ORDER_LINE91CKC;
ALTER TABLE ORDER_LINE91 ADD CONSTRAINT
ORDER_LINE91CKC CHECK (OL_W_ID BETWEEN 144001
AND 145600);
SET INTEGRITY FOR ORDER_LINE91 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE92 OFF;
ALTER TABLE ORDER_LINE92 DROP CONSTRAINT
ORDER_LINE92CKC;
ALTER TABLE ORDER_LINE92 ADD CONSTRAINT
ORDER_LINE92CKC CHECK (OL_W_ID BETWEEN 145601
AND 147200);
SET INTEGRITY FOR ORDER_LINE92 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE93 OFF;
ALTER TABLE ORDER_LINE93 DROP CONSTRAINT
ORDER_LINE93CKC;
ALTER TABLE ORDER_LINE93 ADD CONSTRAINT
ORDER_LINE93CKC CHECK (OL_W_ID BETWEEN 147201
AND 148800);
SET INTEGRITY FOR ORDER_LINE93 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE94 OFF;
ALTER TABLE ORDER_LINE94 DROP CONSTRAINT
ORDER_LINE94CKC;

```

```

ALTER TABLE ORDER_LINE94 ADD CONSTRAINT
ORDER_LINE94CKC CHECK (OL_W_ID BETWEEN 148801
AND 150400);
SET INTEGRITY FOR ORDER_LINE94 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE95 OFF;
ALTER TABLE ORDER_LINE95 DROP CONSTRAINT
ORDER_LINE95CKC;
ALTER TABLE ORDER_LINE95 ADD CONSTRAINT
ORDER_LINE95CKC CHECK (OL_W_ID BETWEEN 150401
AND 152000);
SET INTEGRITY FOR ORDER_LINE95 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE96 OFF;
ALTER TABLE ORDER_LINE96 DROP CONSTRAINT
ORDER_LINE96CKC;
ALTER TABLE ORDER_LINE96 ADD CONSTRAINT
ORDER_LINE96CKC CHECK (OL_W_ID BETWEEN 152001
AND 153600);
SET INTEGRITY FOR ORDER_LINE96 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE97 OFF;
ALTER TABLE ORDER_LINE97 DROP CONSTRAINT
ORDER_LINE97CKC;
ALTER TABLE ORDER_LINE97 ADD CONSTRAINT
ORDER_LINE97CKC CHECK (OL_W_ID BETWEEN 153601
AND 155200);
SET INTEGRITY FOR ORDER_LINE97 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE98 OFF;
ALTER TABLE ORDER_LINE98 DROP CONSTRAINT
ORDER_LINE98CKC;
ALTER TABLE ORDER_LINE98 ADD CONSTRAINT
ORDER_LINE98CKC CHECK (OL_W_ID BETWEEN 155201
AND 156800);
SET INTEGRITY FOR ORDER_LINE98 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE99 OFF;
ALTER TABLE ORDER_LINE99 DROP CONSTRAINT
ORDER_LINE99CKC;
ALTER TABLE ORDER_LINE99 ADD CONSTRAINT
ORDER_LINE99CKC CHECK (OL_W_ID BETWEEN 156801
AND 158400);
SET INTEGRITY FOR ORDER_LINE99 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE100 OFF;
ALTER TABLE ORDER_LINE100 DROP CONSTRAINT
ORDER_LINE100CKC;

```



```

ALTER TABLE ORDER_LINE100 ADD CONSTRAINT
ORDER_LINE100CKC CHECK (OL_W_ID BETWEEN 158401
AND 160000);
SET INTEGRITY FOR ORDER_LINE100 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE101 OFF;
ALTER TABLE ORDER_LINE101 DROP CONSTRAINT
ORDER_LINE101CKC;
ALTER TABLE ORDER_LINE101 ADD CONSTRAINT
ORDER_LINE101CKC CHECK (OL_W_ID BETWEEN 160001
AND 161600);
SET INTEGRITY FOR ORDER_LINE101 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE102 OFF;
ALTER TABLE ORDER_LINE102 DROP CONSTRAINT
ORDER_LINE102CKC;
ALTER TABLE ORDER_LINE102 ADD CONSTRAINT
ORDER_LINE102CKC CHECK (OL_W_ID BETWEEN 161601
AND 163200);
SET INTEGRITY FOR ORDER_LINE102 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE103 OFF;
ALTER TABLE ORDER_LINE103 DROP CONSTRAINT
ORDER_LINE103CKC;
ALTER TABLE ORDER_LINE103 ADD CONSTRAINT
ORDER_LINE103CKC CHECK (OL_W_ID BETWEEN 163201
AND 164800);
SET INTEGRITY FOR ORDER_LINE103 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE104 OFF;
ALTER TABLE ORDER_LINE104 DROP CONSTRAINT
ORDER_LINE104CKC;
ALTER TABLE ORDER_LINE104 ADD CONSTRAINT
ORDER_LINE104CKC CHECK (OL_W_ID BETWEEN 164801
AND 166400);
SET INTEGRITY FOR ORDER_LINE104 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE105 OFF;
ALTER TABLE ORDER_LINE105 DROP CONSTRAINT
ORDER_LINE105CKC;
ALTER TABLE ORDER_LINE105 ADD CONSTRAINT
ORDER_LINE105CKC CHECK (OL_W_ID BETWEEN 166401
AND 168000);
SET INTEGRITY FOR ORDER_LINE105 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE106 OFF;
ALTER TABLE ORDER_LINE106 DROP CONSTRAINT
ORDER_LINE106CKC;

```

```

ALTER TABLE ORDER_LINE106 ADD CONSTRAINT
ORDER_LINE106CKC CHECK (OL_W_ID BETWEEN 168001
AND 169600);
SET INTEGRITY FOR ORDER_LINE106 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE107 OFF;
ALTER TABLE ORDER_LINE107 DROP CONSTRAINT
ORDER_LINE107CKC;
ALTER TABLE ORDER_LINE107 ADD CONSTRAINT
ORDER_LINE107CKC CHECK (OL_W_ID BETWEEN 169601
AND 171200);
SET INTEGRITY FOR ORDER_LINE107 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE108 OFF;
ALTER TABLE ORDER_LINE108 DROP CONSTRAINT
ORDER_LINE108CKC;
ALTER TABLE ORDER_LINE108 ADD CONSTRAINT
ORDER_LINE108CKC CHECK (OL_W_ID BETWEEN 171201
AND 172800);
SET INTEGRITY FOR ORDER_LINE108 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE109 OFF;
ALTER TABLE ORDER_LINE109 DROP CONSTRAINT
ORDER_LINE109CKC;
ALTER TABLE ORDER_LINE109 ADD CONSTRAINT
ORDER_LINE109CKC CHECK (OL_W_ID BETWEEN 172801
AND 174400);
SET INTEGRITY FOR ORDER_LINE109 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE110 OFF;
ALTER TABLE ORDER_LINE110 DROP CONSTRAINT
ORDER_LINE110CKC;
ALTER TABLE ORDER_LINE110 ADD CONSTRAINT
ORDER_LINE110CKC CHECK (OL_W_ID BETWEEN 174401
AND 176000);
SET INTEGRITY FOR ORDER_LINE110 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE111 OFF;
ALTER TABLE ORDER_LINE111 DROP CONSTRAINT
ORDER_LINE111CKC;
ALTER TABLE ORDER_LINE111 ADD CONSTRAINT
ORDER_LINE111CKC CHECK (OL_W_ID BETWEEN 176001
AND 177600);
SET INTEGRITY FOR ORDER_LINE111 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE112 OFF;
ALTER TABLE ORDER_LINE112 DROP CONSTRAINT
ORDER_LINE112CKC;

```

```

ALTER TABLE ORDER_LINE112 ADD CONSTRAINT
ORDER_LINE112CKC CHECK (OL_W_ID BETWEEN 177601
AND 179200);
SET INTEGRITY FOR ORDER_LINE112 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE113 OFF;
ALTER TABLE ORDER_LINE113 DROP CONSTRAINT
ORDER_LINE113CKC;
ALTER TABLE ORDER_LINE113 ADD CONSTRAINT
ORDER_LINE113CKC CHECK (OL_W_ID BETWEEN 179201
AND 180800);
SET INTEGRITY FOR ORDER_LINE113 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE114 OFF;
ALTER TABLE ORDER_LINE114 DROP CONSTRAINT
ORDER_LINE114CKC;
ALTER TABLE ORDER_LINE114 ADD CONSTRAINT
ORDER_LINE114CKC CHECK (OL_W_ID BETWEEN 180801
AND 182400);
SET INTEGRITY FOR ORDER_LINE114 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE115 OFF;
ALTER TABLE ORDER_LINE115 DROP CONSTRAINT
ORDER_LINE115CKC;
ALTER TABLE ORDER_LINE115 ADD CONSTRAINT
ORDER_LINE115CKC CHECK (OL_W_ID BETWEEN 182401
AND 184000);
SET INTEGRITY FOR ORDER_LINE115 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE116 OFF;
ALTER TABLE ORDER_LINE116 DROP CONSTRAINT
ORDER_LINE116CKC;
ALTER TABLE ORDER_LINE116 ADD CONSTRAINT
ORDER_LINE116CKC CHECK (OL_W_ID BETWEEN 184001
AND 185600);
SET INTEGRITY FOR ORDER_LINE116 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE117 OFF;
ALTER TABLE ORDER_LINE117 DROP CONSTRAINT
ORDER_LINE117CKC;
ALTER TABLE ORDER_LINE117 ADD CONSTRAINT
ORDER_LINE117CKC CHECK (OL_W_ID BETWEEN 185601
AND 187200);
SET INTEGRITY FOR ORDER_LINE117 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE118 OFF;
ALTER TABLE ORDER_LINE118 DROP CONSTRAINT
ORDER_LINE118CKC;

```

```

ALTER TABLE ORDER_LINE118 ADD CONSTRAINT
ORDER_LINE118CKC CHECK (OL_W_ID BETWEEN 187201
AND 188800);
SET INTEGRITY FOR ORDER_LINE118 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE119 OFF;
ALTER TABLE ORDER_LINE119 DROP CONSTRAINT
ORDER_LINE119CKC;
ALTER TABLE ORDER_LINE119 ADD CONSTRAINT
ORDER_LINE119CKC CHECK (OL_W_ID BETWEEN 188801
AND 190400);
SET INTEGRITY FOR ORDER_LINE119 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE120 OFF;
ALTER TABLE ORDER_LINE120 DROP CONSTRAINT
ORDER_LINE120CKC;
ALTER TABLE ORDER_LINE120 ADD CONSTRAINT
ORDER_LINE120CKC CHECK (OL_W_ID BETWEEN 190401
AND 192000);
SET INTEGRITY FOR ORDER_LINE120 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE121 OFF;
ALTER TABLE ORDER_LINE121 DROP CONSTRAINT
ORDER_LINE121CKC;
ALTER TABLE ORDER_LINE121 ADD CONSTRAINT
ORDER_LINE121CKC CHECK (OL_W_ID BETWEEN 192001
AND 193600);
SET INTEGRITY FOR ORDER_LINE121 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE122 OFF;
ALTER TABLE ORDER_LINE122 DROP CONSTRAINT
ORDER_LINE122CKC;
ALTER TABLE ORDER_LINE122 ADD CONSTRAINT
ORDER_LINE122CKC CHECK (OL_W_ID BETWEEN 193601
AND 195200);
SET INTEGRITY FOR ORDER_LINE122 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE123 OFF;
ALTER TABLE ORDER_LINE123 DROP CONSTRAINT
ORDER_LINE123CKC;
ALTER TABLE ORDER_LINE123 ADD CONSTRAINT
ORDER_LINE123CKC CHECK (OL_W_ID BETWEEN 195201
AND 196800);
SET INTEGRITY FOR ORDER_LINE123 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE124 OFF;
ALTER TABLE ORDER_LINE124 DROP CONSTRAINT
ORDER_LINE124CKC;

```

```

ALTER TABLE ORDER_LINE124 ADD CONSTRAINT
ORDER_LINE124CKC CHECK (OL_W_ID BETWEEN 196801
AND 198400);
SET INTEGRITY FOR ORDER_LINE124 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE125 OFF;
ALTER TABLE ORDER_LINE125 DROP CONSTRAINT
ORDER_LINE125CKC;
ALTER TABLE ORDER_LINE125 ADD CONSTRAINT
ORDER_LINE125CKC CHECK (OL_W_ID BETWEEN 198401
AND 200000);
SET INTEGRITY FOR ORDER_LINE125 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE126 OFF;
ALTER TABLE ORDER_LINE126 DROP CONSTRAINT
ORDER_LINE126CKC;
ALTER TABLE ORDER_LINE126 ADD CONSTRAINT
ORDER_LINE126CKC CHECK (OL_W_ID BETWEEN 200001
AND 201600);
SET INTEGRITY FOR ORDER_LINE126 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE127 OFF;
ALTER TABLE ORDER_LINE127 DROP CONSTRAINT
ORDER_LINE127CKC;
ALTER TABLE ORDER_LINE127 ADD CONSTRAINT
ORDER_LINE127CKC CHECK (OL_W_ID BETWEEN 201601
AND 203200);
SET INTEGRITY FOR ORDER_LINE127 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE128 OFF;
ALTER TABLE ORDER_LINE128 DROP CONSTRAINT
ORDER_LINE128CKC;
ALTER TABLE ORDER_LINE128 ADD CONSTRAINT
ORDER_LINE128CKC CHECK (OL_W_ID BETWEEN 203201
AND 204800);
SET INTEGRITY FOR ORDER_LINE128 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE129 OFF;
ALTER TABLE ORDER_LINE129 DROP CONSTRAINT
ORDER_LINE129CKC;
ALTER TABLE ORDER_LINE129 ADD CONSTRAINT
ORDER_LINE129CKC CHECK (OL_W_ID BETWEEN 204801
AND 206400);
SET INTEGRITY FOR ORDER_LINE129 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE130 OFF;
ALTER TABLE ORDER_LINE130 DROP CONSTRAINT
ORDER_LINE130CKC;

```

```

ALTER TABLE ORDER_LINE130 ADD CONSTRAINT
ORDER_LINE130CKC CHECK (OL_W_ID BETWEEN 206401
AND 208000);
SET INTEGRITY FOR ORDER_LINE130 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE131 OFF;
ALTER TABLE ORDER_LINE131 DROP CONSTRAINT
ORDER_LINE131CKC;
ALTER TABLE ORDER_LINE131 ADD CONSTRAINT
ORDER_LINE131CKC CHECK (OL_W_ID BETWEEN 208001
AND 209600);
SET INTEGRITY FOR ORDER_LINE131 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE132 OFF;
ALTER TABLE ORDER_LINE132 DROP CONSTRAINT
ORDER_LINE132CKC;
ALTER TABLE ORDER_LINE132 ADD CONSTRAINT
ORDER_LINE132CKC CHECK (OL_W_ID BETWEEN 209601
AND 211200);
SET INTEGRITY FOR ORDER_LINE132 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE133 OFF;
ALTER TABLE ORDER_LINE133 DROP CONSTRAINT
ORDER_LINE133CKC;
ALTER TABLE ORDER_LINE133 ADD CONSTRAINT
ORDER_LINE133CKC CHECK (OL_W_ID BETWEEN 211201
AND 212800);
SET INTEGRITY FOR ORDER_LINE133 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE134 OFF;
ALTER TABLE ORDER_LINE134 DROP CONSTRAINT
ORDER_LINE134CKC;
ALTER TABLE ORDER_LINE134 ADD CONSTRAINT
ORDER_LINE134CKC CHECK (OL_W_ID BETWEEN 212801
AND 214400);
SET INTEGRITY FOR ORDER_LINE134 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE135 OFF;
ALTER TABLE ORDER_LINE135 DROP CONSTRAINT
ORDER_LINE135CKC;
ALTER TABLE ORDER_LINE135 ADD CONSTRAINT
ORDER_LINE135CKC CHECK (OL_W_ID BETWEEN 214401
AND 216000);
SET INTEGRITY FOR ORDER_LINE135 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE136 OFF;
ALTER TABLE ORDER_LINE136 DROP CONSTRAINT
ORDER_LINE136CKC;

```

```

ALTER TABLE ORDER_LINE136 ADD CONSTRAINT
ORDER_LINE136CKC CHECK (OL_W_ID BETWEEN 216001
AND 217600);
SET INTEGRITY FOR ORDER_LINE136 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE137 OFF;
ALTER TABLE ORDER_LINE137 DROP CONSTRAINT
ORDER_LINE137CKC;
ALTER TABLE ORDER_LINE137 ADD CONSTRAINT
ORDER_LINE137CKC CHECK (OL_W_ID BETWEEN 217601
AND 219200);
SET INTEGRITY FOR ORDER_LINE137 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE138 OFF;
ALTER TABLE ORDER_LINE138 DROP CONSTRAINT
ORDER_LINE138CKC;
ALTER TABLE ORDER_LINE138 ADD CONSTRAINT
ORDER_LINE138CKC CHECK (OL_W_ID BETWEEN 219201
AND 220800);
SET INTEGRITY FOR ORDER_LINE138 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE139 OFF;
ALTER TABLE ORDER_LINE139 DROP CONSTRAINT
ORDER_LINE139CKC;
ALTER TABLE ORDER_LINE139 ADD CONSTRAINT
ORDER_LINE139CKC CHECK (OL_W_ID BETWEEN 220801
AND 222400);
SET INTEGRITY FOR ORDER_LINE139 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE140 OFF;
ALTER TABLE ORDER_LINE140 DROP CONSTRAINT
ORDER_LINE140CKC;
ALTER TABLE ORDER_LINE140 ADD CONSTRAINT
ORDER_LINE140CKC CHECK (OL_W_ID BETWEEN 222401
AND 224000);
SET INTEGRITY FOR ORDER_LINE140 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE141 OFF;
ALTER TABLE ORDER_LINE141 DROP CONSTRAINT
ORDER_LINE141CKC;
ALTER TABLE ORDER_LINE141 ADD CONSTRAINT
ORDER_LINE141CKC CHECK (OL_W_ID BETWEEN 224001
AND 225600);
SET INTEGRITY FOR ORDER_LINE141 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE142 OFF;
ALTER TABLE ORDER_LINE142 DROP CONSTRAINT
ORDER_LINE142CKC;

```

```

ALTER TABLE ORDER_LINE142 ADD CONSTRAINT
ORDER_LINE142CKC CHECK (OL_W_ID BETWEEN 225601
AND 227200);
SET INTEGRITY FOR ORDER_LINE142 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE143 OFF;
ALTER TABLE ORDER_LINE143 DROP CONSTRAINT
ORDER_LINE143CKC;
ALTER TABLE ORDER_LINE143 ADD CONSTRAINT
ORDER_LINE143CKC CHECK (OL_W_ID BETWEEN 227201
AND 228800);
SET INTEGRITY FOR ORDER_LINE143 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE144 OFF;
ALTER TABLE ORDER_LINE144 DROP CONSTRAINT
ORDER_LINE144CKC;
ALTER TABLE ORDER_LINE144 ADD CONSTRAINT
ORDER_LINE144CKC CHECK (OL_W_ID BETWEEN 228801
AND 230400);
SET INTEGRITY FOR ORDER_LINE144 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE145 OFF;
ALTER TABLE ORDER_LINE145 DROP CONSTRAINT
ORDER_LINE145CKC;
ALTER TABLE ORDER_LINE145 ADD CONSTRAINT
ORDER_LINE145CKC CHECK (OL_W_ID BETWEEN 230401
AND 232000);
SET INTEGRITY FOR ORDER_LINE145 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE146 OFF;
ALTER TABLE ORDER_LINE146 DROP CONSTRAINT
ORDER_LINE146CKC;
ALTER TABLE ORDER_LINE146 ADD CONSTRAINT
ORDER_LINE146CKC CHECK (OL_W_ID BETWEEN 232001
AND 233600);
SET INTEGRITY FOR ORDER_LINE146 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE147 OFF;
ALTER TABLE ORDER_LINE147 DROP CONSTRAINT
ORDER_LINE147CKC;
ALTER TABLE ORDER_LINE147 ADD CONSTRAINT
ORDER_LINE147CKC CHECK (OL_W_ID BETWEEN 233601
AND 235200);
SET INTEGRITY FOR ORDER_LINE147 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE148 OFF;
ALTER TABLE ORDER_LINE148 DROP CONSTRAINT
ORDER_LINE148CKC;

```

```

ALTER TABLE ORDER_LINE148 ADD CONSTRAINT
ORDER_LINE148CKC CHECK (OL_W_ID BETWEEN 235201
AND 236800);
SET INTEGRITY FOR ORDER_LINE148 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE149 OFF;
ALTER TABLE ORDER_LINE149 DROP CONSTRAINT
ORDER_LINE149CKC;
ALTER TABLE ORDER_LINE149 ADD CONSTRAINT
ORDER_LINE149CKC CHECK (OL_W_ID BETWEEN 236801
AND 238400);
SET INTEGRITY FOR ORDER_LINE149 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE150 OFF;
ALTER TABLE ORDER_LINE150 DROP CONSTRAINT
ORDER_LINE150CKC;
ALTER TABLE ORDER_LINE150 ADD CONSTRAINT
ORDER_LINE150CKC CHECK (OL_W_ID BETWEEN 238401
AND 240000);
SET INTEGRITY FOR ORDER_LINE150 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE151 OFF;
ALTER TABLE ORDER_LINE151 DROP CONSTRAINT
ORDER_LINE151CKC;
ALTER TABLE ORDER_LINE151 ADD CONSTRAINT
ORDER_LINE151CKC CHECK (OL_W_ID BETWEEN 240001
AND 241600);
SET INTEGRITY FOR ORDER_LINE151 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE152 OFF;
ALTER TABLE ORDER_LINE152 DROP CONSTRAINT
ORDER_LINE152CKC;
ALTER TABLE ORDER_LINE152 ADD CONSTRAINT
ORDER_LINE152CKC CHECK (OL_W_ID BETWEEN 241601
AND 243200);
SET INTEGRITY FOR ORDER_LINE152 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE153 OFF;
ALTER TABLE ORDER_LINE153 DROP CONSTRAINT
ORDER_LINE153CKC;
ALTER TABLE ORDER_LINE153 ADD CONSTRAINT
ORDER_LINE153CKC CHECK (OL_W_ID BETWEEN 243201
AND 244800);
SET INTEGRITY FOR ORDER_LINE153 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE154 OFF;
ALTER TABLE ORDER_LINE154 DROP CONSTRAINT
ORDER_LINE154CKC;

```

```

ALTER TABLE ORDER_LINE154 ADD CONSTRAINT
ORDER_LINE154CKC CHECK (OL_W_ID BETWEEN 244801
AND 246400);
SET INTEGRITY FOR ORDER_LINE154 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE155 OFF;
ALTER TABLE ORDER_LINE155 DROP CONSTRAINT
ORDER_LINE155CKC;
ALTER TABLE ORDER_LINE155 ADD CONSTRAINT
ORDER_LINE155CKC CHECK (OL_W_ID BETWEEN 246401
AND 248000);
SET INTEGRITY FOR ORDER_LINE155 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE156 OFF;
ALTER TABLE ORDER_LINE156 DROP CONSTRAINT
ORDER_LINE156CKC;
ALTER TABLE ORDER_LINE156 ADD CONSTRAINT
ORDER_LINE156CKC CHECK (OL_W_ID BETWEEN 248001
AND 249600);
SET INTEGRITY FOR ORDER_LINE156 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE157 OFF;
ALTER TABLE ORDER_LINE157 DROP CONSTRAINT
ORDER_LINE157CKC;
ALTER TABLE ORDER_LINE157 ADD CONSTRAINT
ORDER_LINE157CKC CHECK (OL_W_ID BETWEEN 249601
AND 251200);
SET INTEGRITY FOR ORDER_LINE157 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE158 OFF;
ALTER TABLE ORDER_LINE158 DROP CONSTRAINT
ORDER_LINE158CKC;
ALTER TABLE ORDER_LINE158 ADD CONSTRAINT
ORDER_LINE158CKC CHECK (OL_W_ID BETWEEN 251201
AND 252800);
SET INTEGRITY FOR ORDER_LINE158 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE159 OFF;
ALTER TABLE ORDER_LINE159 DROP CONSTRAINT
ORDER_LINE159CKC;
ALTER TABLE ORDER_LINE159 ADD CONSTRAINT
ORDER_LINE159CKC CHECK (OL_W_ID BETWEEN 252801
AND 254400);
SET INTEGRITY FOR ORDER_LINE159 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE160 OFF;
ALTER TABLE ORDER_LINE160 DROP CONSTRAINT
ORDER_LINE160CKC;
ALTER TABLE ORDER_LINE160 ADD CONSTRAINT
ORDER_LINE160CKC CHECK (OL_W_ID >= 254401);

```

```

SET INTEGRITY FOR ORDER_LINE160 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRCONST STOCK.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK1 OFF;
ALTER TABLE STOCK1 DROP CONSTRAINT STOCK1CKC;
ALTER TABLE STOCK1 ADD CONSTRAINT STOCK1CKC
CHECK (S_W_ID BETWEEN 1 AND 1600);
SET INTEGRITY FOR STOCK1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK2 OFF;
ALTER TABLE STOCK2 DROP CONSTRAINT STOCK2CKC;
ALTER TABLE STOCK2 ADD CONSTRAINT STOCK2CKC
CHECK (S_W_ID BETWEEN 1601 AND 3200);
SET INTEGRITY FOR STOCK2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK3 OFF;
ALTER TABLE STOCK3 DROP CONSTRAINT STOCK3CKC;
ALTER TABLE STOCK3 ADD CONSTRAINT STOCK3CKC
CHECK (S_W_ID BETWEEN 3201 AND 4800);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK4 OFF;
ALTER TABLE STOCK4 DROP CONSTRAINT STOCK4CKC;
ALTER TABLE STOCK4 ADD CONSTRAINT STOCK4CKC
CHECK (S_W_ID BETWEEN 4801 AND 6400);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK5 OFF;
ALTER TABLE STOCK5 DROP CONSTRAINT STOCK5CKC;
ALTER TABLE STOCK5 ADD CONSTRAINT STOCK5CKC
CHECK (S_W_ID BETWEEN 6401 AND 8000);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK6 OFF;
ALTER TABLE STOCK6 DROP CONSTRAINT STOCK6CKC;
ALTER TABLE STOCK6 ADD CONSTRAINT STOCK6CKC
CHECK (S_W_ID BETWEEN 8001 AND 9600);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK7 OFF;
ALTER TABLE STOCK7 DROP CONSTRAINT STOCK7CKC;
ALTER TABLE STOCK7 ADD CONSTRAINT STOCK7CKC
CHECK (S_W_ID BETWEEN 9601 AND 11200);
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 11201 AND 12800);
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 12801 AND 14400);
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 14401 AND 16000);
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 16001 AND 17600);
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 17601 AND 19200);
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 19201 AND 20800);
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 20801 AND 22400);
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 22401 AND 24000);
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 24001 AND 25600);
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 25601 AND 27200);
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 27201 AND 28800);
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 28801 AND 30400);
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 30401 AND 32000);
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 32001 AND 33600);
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 33601 AND 35200);
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 35201 AND 36800);
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 36801 AND 38400);
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 38401 AND 40000);
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 40001 AND 41600);
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 41601 AND 43200);
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 43201 AND 44800);

```

```

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 44801 AND 46400);
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 46401 AND 48000);
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 48001 AND 49600);
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 49601 AND 51200);
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 51201 AND 52800);
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 52801 AND 54400);
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 54401 AND 56000);
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 56001 AND 57600);
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 57601 AND 59200);
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 59201 AND 60800);
SET INTEGRITY FOR STOCK38 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK39 OFF;
ALTER TABLE STOCK39 DROP CONSTRAINT
STOCK39CKC;
ALTER TABLE STOCK39 ADD CONSTRAINT STOCK39CKC
CHECK (S_W_ID BETWEEN 60801 AND 62400);
SET INTEGRITY FOR STOCK39 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK40 OFF;
ALTER TABLE STOCK40 DROP CONSTRAINT
STOCK40CKC;
ALTER TABLE STOCK40 ADD CONSTRAINT STOCK40CKC
CHECK (S_W_ID BETWEEN 62401 AND 64000);
SET INTEGRITY FOR STOCK40 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK41 OFF;
ALTER TABLE STOCK41 DROP CONSTRAINT
STOCK41CKC;
ALTER TABLE STOCK41 ADD CONSTRAINT STOCK41CKC
CHECK (S_W_ID BETWEEN 64001 AND 65600);
SET INTEGRITY FOR STOCK41 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK42 OFF;

```

```

ALTER TABLE STOCK42 DROP CONSTRAINT
STOCK42CKC;
ALTER TABLE STOCK42 ADD CONSTRAINT STOCK42CKC
CHECK (S_W_ID BETWEEN 65601 AND 67200);
SET INTEGRITY FOR STOCK42 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK43 OFF;
ALTER TABLE STOCK43 DROP CONSTRAINT
STOCK43CKC;
ALTER TABLE STOCK43 ADD CONSTRAINT STOCK43CKC
CHECK (S_W_ID BETWEEN 67201 AND 68800);
SET INTEGRITY FOR STOCK43 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK44 OFF;
ALTER TABLE STOCK44 DROP CONSTRAINT
STOCK44CKC;
ALTER TABLE STOCK44 ADD CONSTRAINT STOCK44CKC
CHECK (S_W_ID BETWEEN 68801 AND 70400);
SET INTEGRITY FOR STOCK44 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK45 OFF;
ALTER TABLE STOCK45 DROP CONSTRAINT
STOCK45CKC;
ALTER TABLE STOCK45 ADD CONSTRAINT STOCK45CKC
CHECK (S_W_ID BETWEEN 70401 AND 72000);
SET INTEGRITY FOR STOCK45 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK46 OFF;
ALTER TABLE STOCK46 DROP CONSTRAINT
STOCK46CKC;
ALTER TABLE STOCK46 ADD CONSTRAINT STOCK46CKC
CHECK (S_W_ID BETWEEN 72001 AND 73600);
SET INTEGRITY FOR STOCK46 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK47 OFF;
ALTER TABLE STOCK47 DROP CONSTRAINT
STOCK47CKC;
ALTER TABLE STOCK47 ADD CONSTRAINT STOCK47CKC
CHECK (S_W_ID BETWEEN 73601 AND 75200);
SET INTEGRITY FOR STOCK47 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK48 OFF;
ALTER TABLE STOCK48 DROP CONSTRAINT
STOCK48CKC;
ALTER TABLE STOCK48 ADD CONSTRAINT STOCK48CKC
CHECK (S_W_ID BETWEEN 75201 AND 76800);
SET INTEGRITY FOR STOCK48 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;

```

```

SET INTEGRITY FOR STOCK49 OFF;
ALTER TABLE STOCK49 DROP CONSTRAINT
STOCK49CKC;
ALTER TABLE STOCK49 ADD CONSTRAINT STOCK49CKC
CHECK (S_W_ID BETWEEN 76801 AND 78400);
SET INTEGRITY FOR STOCK49 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK50 OFF;
ALTER TABLE STOCK50 DROP CONSTRAINT
STOCK50CKC;
ALTER TABLE STOCK50 ADD CONSTRAINT STOCK50CKC
CHECK (S_W_ID BETWEEN 78401 AND 80000);
SET INTEGRITY FOR STOCK50 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK51 OFF;
ALTER TABLE STOCK51 DROP CONSTRAINT
STOCK51CKC;
ALTER TABLE STOCK51 ADD CONSTRAINT STOCK51CKC
CHECK (S_W_ID BETWEEN 80001 AND 81600);
SET INTEGRITY FOR STOCK51 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK52 OFF;
ALTER TABLE STOCK52 DROP CONSTRAINT
STOCK52CKC;
ALTER TABLE STOCK52 ADD CONSTRAINT STOCK52CKC
CHECK (S_W_ID BETWEEN 81601 AND 83200);
SET INTEGRITY FOR STOCK52 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK53 OFF;
ALTER TABLE STOCK53 DROP CONSTRAINT
STOCK53CKC;
ALTER TABLE STOCK53 ADD CONSTRAINT STOCK53CKC
CHECK (S_W_ID BETWEEN 83201 AND 84800);
SET INTEGRITY FOR STOCK53 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK54 OFF;
ALTER TABLE STOCK54 DROP CONSTRAINT
STOCK54CKC;
ALTER TABLE STOCK54 ADD CONSTRAINT STOCK54CKC
CHECK (S_W_ID BETWEEN 84801 AND 86400);
SET INTEGRITY FOR STOCK54 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK55 OFF;
ALTER TABLE STOCK55 DROP CONSTRAINT
STOCK55CKC;
ALTER TABLE STOCK55 ADD CONSTRAINT STOCK55CKC
CHECK (S_W_ID BETWEEN 86401 AND 88000);
SET INTEGRITY FOR STOCK55 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR STOCK56 OFF;
ALTER TABLE STOCK56 DROP CONSTRAINT
STOCK56CKC;
ALTER TABLE STOCK56 ADD CONSTRAINT STOCK56CKC
CHECK (S_W_ID BETWEEN 88001 AND 89600);
SET INTEGRITY FOR STOCK56 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK57 OFF;
ALTER TABLE STOCK57 DROP CONSTRAINT
STOCK57CKC;
ALTER TABLE STOCK57 ADD CONSTRAINT STOCK57CKC
CHECK (S_W_ID BETWEEN 89601 AND 91200);
SET INTEGRITY FOR STOCK57 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK58 OFF;
ALTER TABLE STOCK58 DROP CONSTRAINT
STOCK58CKC;
ALTER TABLE STOCK58 ADD CONSTRAINT STOCK58CKC
CHECK (S_W_ID BETWEEN 91201 AND 92800);
SET INTEGRITY FOR STOCK58 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK59 OFF;
ALTER TABLE STOCK59 DROP CONSTRAINT
STOCK59CKC;
ALTER TABLE STOCK59 ADD CONSTRAINT STOCK59CKC
CHECK (S_W_ID BETWEEN 92801 AND 94400);
SET INTEGRITY FOR STOCK59 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK60 OFF;
ALTER TABLE STOCK60 DROP CONSTRAINT
STOCK60CKC;
ALTER TABLE STOCK60 ADD CONSTRAINT STOCK60CKC
CHECK (S_W_ID BETWEEN 94401 AND 96000);
SET INTEGRITY FOR STOCK60 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK61 OFF;
ALTER TABLE STOCK61 DROP CONSTRAINT
STOCK61CKC;
ALTER TABLE STOCK61 ADD CONSTRAINT STOCK61CKC
CHECK (S_W_ID BETWEEN 96001 AND 97600);
SET INTEGRITY FOR STOCK61 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK62 OFF;
ALTER TABLE STOCK62 DROP CONSTRAINT
STOCK62CKC;
ALTER TABLE STOCK62 ADD CONSTRAINT STOCK62CKC
CHECK (S_W_ID BETWEEN 97601 AND 99200);
SET INTEGRITY FOR STOCK62 ALL IMMEDIATE
UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK63 OFF;
ALTER TABLE STOCK63 DROP CONSTRAINT
STOCK63CKC;
ALTER TABLE STOCK63 ADD CONSTRAINT STOCK63CKC
CHECK (S_W_ID BETWEEN 99201 AND 100800);
SET INTEGRITY FOR STOCK63 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK64 OFF;
ALTER TABLE STOCK64 DROP CONSTRAINT
STOCK64CKC;
ALTER TABLE STOCK64 ADD CONSTRAINT STOCK64CKC
CHECK (S_W_ID BETWEEN 100801 AND 102400);
SET INTEGRITY FOR STOCK64 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK65 OFF;
ALTER TABLE STOCK65 DROP CONSTRAINT
STOCK65CKC;
ALTER TABLE STOCK65 ADD CONSTRAINT STOCK65CKC
CHECK (S_W_ID BETWEEN 102401 AND 104000);
SET INTEGRITY FOR STOCK65 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK66 OFF;
ALTER TABLE STOCK66 DROP CONSTRAINT
STOCK66CKC;
ALTER TABLE STOCK66 ADD CONSTRAINT STOCK66CKC
CHECK (S_W_ID BETWEEN 104001 AND 105600);
SET INTEGRITY FOR STOCK66 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK67 OFF;
ALTER TABLE STOCK67 DROP CONSTRAINT
STOCK67CKC;
ALTER TABLE STOCK67 ADD CONSTRAINT STOCK67CKC
CHECK (S_W_ID BETWEEN 105601 AND 107200);
SET INTEGRITY FOR STOCK67 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK68 OFF;
ALTER TABLE STOCK68 DROP CONSTRAINT
STOCK68CKC;
ALTER TABLE STOCK68 ADD CONSTRAINT STOCK68CKC
CHECK (S_W_ID BETWEEN 107201 AND 108800);
SET INTEGRITY FOR STOCK68 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK69 OFF;
ALTER TABLE STOCK69 DROP CONSTRAINT
STOCK69CKC;
ALTER TABLE STOCK69 ADD CONSTRAINT STOCK69CKC
CHECK (S_W_ID BETWEEN 108801 AND 110400);

```

```

SET INTEGRITY FOR STOCK69 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK70 OFF;
ALTER TABLE STOCK70 DROP CONSTRAINT
STOCK70CKC;
ALTER TABLE STOCK70 ADD CONSTRAINT STOCK70CKC
CHECK (S_W_ID BETWEEN 110401 AND 112000);
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 112001 AND 113600);
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 113601 AND 115200);
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 115201 AND 116800);
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 116801 AND 118400);
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 118401 AND 120000);
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 120001 AND 121600);
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 121601 AND 123200);
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 123201 AND 124800);
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 124801 AND 126400);
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 126401 AND 128000);
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 128001 AND 129600);
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 129601 AND 131200);
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 131201 AND 132800);
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 132801 AND 134400);
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 134401 AND 136000);
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 136001 AND 137600);
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 137601 AND 139200);
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 139201 AND 140800);
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 140801 AND 142400);
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 142401 AND 144000);
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 144001 AND 145600);
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 145601 AND 147200);
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 147201 AND 148800);
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 148801 AND 150400);
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 150401 AND 152000);
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 152001 AND 153600);
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 153601 AND 155200);
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 155201 AND 156800);
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 156801 AND 158400);
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 158401 AND
160000);
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 160001 AND
161600);
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 161601 AND
163200);
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 163201 AND
164800);
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 164801 AND
166400);
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 166401 AND
168000);
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 168001 AND
169600);
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 169601 AND
171200);
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 171201 AND
172800);
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 172801 AND
174400);
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 174401 AND
176000);
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 176001 AND
177600);
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 177601 AND
179200);
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 179201 AND
180800);
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 180801 AND
182400);
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 182401 AND
184000);
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 184001 AND
185600);
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 185601 AND
187200);
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 187201 AND
188800);
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 188801 AND
190400);
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 BETWEEN 190401 AND
192000);
SET INTEGRITY FOR STOCK120 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK121 OFF;
ALTER TABLE STOCK121 DROP CONSTRAINT
STOCK121CKC;

```

```

ALTER TABLE STOCK121 ADD CONSTRAINT
STOCK121CKC CHECK (S_W_ID BETWEEN 192001 AND
193600);
SET INTEGRITY FOR STOCK121 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK122 OFF;
ALTER TABLE STOCK122 DROP CONSTRAINT
STOCK122CKC;
ALTER TABLE STOCK122 ADD CONSTRAINT
STOCK122CKC CHECK (S_W_ID BETWEEN 193601 AND
195200);
SET INTEGRITY FOR STOCK122 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK123 OFF;
ALTER TABLE STOCK123 DROP CONSTRAINT
STOCK123CKC;
ALTER TABLE STOCK123 ADD CONSTRAINT
STOCK123CKC CHECK (S_W_ID BETWEEN 195201 AND
196800);
SET INTEGRITY FOR STOCK123 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK124 OFF;
ALTER TABLE STOCK124 DROP CONSTRAINT
STOCK124CKC;
ALTER TABLE STOCK124 ADD CONSTRAINT
STOCK124CKC CHECK (S_W_ID BETWEEN 196801 AND
198400);
SET INTEGRITY FOR STOCK124 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK125 OFF;
ALTER TABLE STOCK125 DROP CONSTRAINT
STOCK125CKC;
ALTER TABLE STOCK125 ADD CONSTRAINT
STOCK125CKC CHECK (S_W_ID BETWEEN 198401 AND
200000);
SET INTEGRITY FOR STOCK125 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK126 OFF;
ALTER TABLE STOCK126 DROP CONSTRAINT
STOCK126CKC;
ALTER TABLE STOCK126 ADD CONSTRAINT
STOCK126CKC CHECK (S_W_ID BETWEEN 200001 AND
201600);
SET INTEGRITY FOR STOCK126 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK127 OFF;
ALTER TABLE STOCK127 DROP CONSTRAINT
STOCK127CKC;

```

```

ALTER TABLE STOCK127 ADD CONSTRAINT
STOCK127CKC CHECK (S_W_ID BETWEEN 201601 AND
203200);
SET INTEGRITY FOR STOCK127 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK128 OFF;
ALTER TABLE STOCK128 DROP CONSTRAINT
STOCK128CKC;
ALTER TABLE STOCK128 ADD CONSTRAINT
STOCK128CKC CHECK (S_W_ID BETWEEN 203201 AND
204800);
SET INTEGRITY FOR STOCK128 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK129 OFF;
ALTER TABLE STOCK129 DROP CONSTRAINT
STOCK129CKC;
ALTER TABLE STOCK129 ADD CONSTRAINT
STOCK129CKC CHECK (S_W_ID BETWEEN 204801 AND
206400);
SET INTEGRITY FOR STOCK129 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK130 OFF;
ALTER TABLE STOCK130 DROP CONSTRAINT
STOCK130CKC;
ALTER TABLE STOCK130 ADD CONSTRAINT
STOCK130CKC CHECK (S_W_ID BETWEEN 206401 AND
208000);
SET INTEGRITY FOR STOCK130 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK131 OFF;
ALTER TABLE STOCK131 DROP CONSTRAINT
STOCK131CKC;
ALTER TABLE STOCK131 ADD CONSTRAINT
STOCK131CKC CHECK (S_W_ID BETWEEN 208001 AND
209600);
SET INTEGRITY FOR STOCK131 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK132 OFF;
ALTER TABLE STOCK132 DROP CONSTRAINT
STOCK132CKC;
ALTER TABLE STOCK132 ADD CONSTRAINT
STOCK132CKC CHECK (S_W_ID BETWEEN 209601 AND
211200);
SET INTEGRITY FOR STOCK132 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK133 OFF;
ALTER TABLE STOCK133 DROP CONSTRAINT
STOCK133CKC;

```

```

ALTER TABLE STOCK133 ADD CONSTRAINT
STOCK133CKC CHECK (S_W_ID BETWEEN 211201 AND
212800);
SET INTEGRITY FOR STOCK133 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK134 OFF;
ALTER TABLE STOCK134 DROP CONSTRAINT
STOCK134CKC;
ALTER TABLE STOCK134 ADD CONSTRAINT
STOCK134CKC CHECK (S_W_ID BETWEEN 212801 AND
214400);
SET INTEGRITY FOR STOCK134 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK135 OFF;
ALTER TABLE STOCK135 DROP CONSTRAINT
STOCK135CKC;
ALTER TABLE STOCK135 ADD CONSTRAINT
STOCK135CKC CHECK (S_W_ID BETWEEN 214401 AND
216000);
SET INTEGRITY FOR STOCK135 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK136 OFF;
ALTER TABLE STOCK136 DROP CONSTRAINT
STOCK136CKC;
ALTER TABLE STOCK136 ADD CONSTRAINT
STOCK136CKC CHECK (S_W_ID BETWEEN 216001 AND
217600);
SET INTEGRITY FOR STOCK136 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK137 OFF;
ALTER TABLE STOCK137 DROP CONSTRAINT
STOCK137CKC;
ALTER TABLE STOCK137 ADD CONSTRAINT
STOCK137CKC CHECK (S_W_ID BETWEEN 217601 AND
219200);
SET INTEGRITY FOR STOCK137 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK138 OFF;
ALTER TABLE STOCK138 DROP CONSTRAINT
STOCK138CKC;
ALTER TABLE STOCK138 ADD CONSTRAINT
STOCK138CKC CHECK (S_W_ID BETWEEN 219201 AND
220800);
SET INTEGRITY FOR STOCK138 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK139 OFF;
ALTER TABLE STOCK139 DROP CONSTRAINT
STOCK139CKC;

```

```

ALTER TABLE STOCK139 ADD CONSTRAINT
STOCK139CKC CHECK (S_W_ID BETWEEN 220801 AND
222400);
SET INTEGRITY FOR STOCK139 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK140 OFF;
ALTER TABLE STOCK140 DROP CONSTRAINT
STOCK140CKC;
ALTER TABLE STOCK140 ADD CONSTRAINT
STOCK140CKC CHECK (S_W_ID BETWEEN 222401 AND
224000);
SET INTEGRITY FOR STOCK140 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK141 OFF;
ALTER TABLE STOCK141 DROP CONSTRAINT
STOCK141CKC;
ALTER TABLE STOCK141 ADD CONSTRAINT
STOCK141CKC CHECK (S_W_ID BETWEEN 224001 AND
225600);
SET INTEGRITY FOR STOCK141 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK142 OFF;
ALTER TABLE STOCK142 DROP CONSTRAINT
STOCK142CKC;
ALTER TABLE STOCK142 ADD CONSTRAINT
STOCK142CKC CHECK (S_W_ID BETWEEN 225601 AND
227200);
SET INTEGRITY FOR STOCK142 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK143 OFF;
ALTER TABLE STOCK143 DROP CONSTRAINT
STOCK143CKC;
ALTER TABLE STOCK143 ADD CONSTRAINT
STOCK143CKC CHECK (S_W_ID BETWEEN 227201 AND
228800);
SET INTEGRITY FOR STOCK143 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK144 OFF;
ALTER TABLE STOCK144 DROP CONSTRAINT
STOCK144CKC;
ALTER TABLE STOCK144 ADD CONSTRAINT
STOCK144CKC CHECK (S_W_ID BETWEEN 228801 AND
230400);
SET INTEGRITY FOR STOCK144 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK145 OFF;
ALTER TABLE STOCK145 DROP CONSTRAINT
STOCK145CKC;

```

```

ALTER TABLE STOCK145 ADD CONSTRAINT
STOCK145CKC CHECK (S_W_ID BETWEEN 230401 AND
232000);
SET INTEGRITY FOR STOCK145 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK146 OFF;
ALTER TABLE STOCK146 DROP CONSTRAINT
STOCK146CKC;
ALTER TABLE STOCK146 ADD CONSTRAINT
STOCK146CKC CHECK (S_W_ID BETWEEN 232001 AND
233600);
SET INTEGRITY FOR STOCK146 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK147 OFF;
ALTER TABLE STOCK147 DROP CONSTRAINT
STOCK147CKC;
ALTER TABLE STOCK147 ADD CONSTRAINT
STOCK147CKC CHECK (S_W_ID BETWEEN 233601 AND
235200);
SET INTEGRITY FOR STOCK147 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK148 OFF;
ALTER TABLE STOCK148 DROP CONSTRAINT
STOCK148CKC;
ALTER TABLE STOCK148 ADD CONSTRAINT
STOCK148CKC CHECK (S_W_ID BETWEEN 235201 AND
236800);
SET INTEGRITY FOR STOCK148 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK149 OFF;
ALTER TABLE STOCK149 DROP CONSTRAINT
STOCK149CKC;
ALTER TABLE STOCK149 ADD CONSTRAINT
STOCK149CKC CHECK (S_W_ID BETWEEN 236801 AND
238400);
SET INTEGRITY FOR STOCK149 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK150 OFF;
ALTER TABLE STOCK150 DROP CONSTRAINT
STOCK150CKC;
ALTER TABLE STOCK150 ADD CONSTRAINT
STOCK150CKC CHECK (S_W_ID BETWEEN 238401 AND
240000);
SET INTEGRITY FOR STOCK150 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK151 OFF;
ALTER TABLE STOCK151 DROP CONSTRAINT
STOCK151CKC;

```

```

ALTER TABLE STOCK151 ADD CONSTRAINT
STOCK151CKC CHECK (S_W_ID BETWEEN 240001 AND
241600);
SET INTEGRITY FOR STOCK151 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK152 OFF;
ALTER TABLE STOCK152 DROP CONSTRAINT
STOCK152CKC;
ALTER TABLE STOCK152 ADD CONSTRAINT
STOCK152CKC CHECK (S_W_ID BETWEEN 241601 AND
243200);
SET INTEGRITY FOR STOCK152 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK153 OFF;
ALTER TABLE STOCK153 DROP CONSTRAINT
STOCK153CKC;
ALTER TABLE STOCK153 ADD CONSTRAINT
STOCK153CKC CHECK (S_W_ID BETWEEN 243201 AND
244800);
SET INTEGRITY FOR STOCK153 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK154 OFF;
ALTER TABLE STOCK154 DROP CONSTRAINT
STOCK154CKC;
ALTER TABLE STOCK154 ADD CONSTRAINT
STOCK154CKC CHECK (S_W_ID BETWEEN 244801 AND
246400);
SET INTEGRITY FOR STOCK154 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK155 OFF;
ALTER TABLE STOCK155 DROP CONSTRAINT
STOCK155CKC;
ALTER TABLE STOCK155 ADD CONSTRAINT
STOCK155CKC CHECK (S_W_ID BETWEEN 246401 AND
248000);
SET INTEGRITY FOR STOCK155 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK156 OFF;
ALTER TABLE STOCK156 DROP CONSTRAINT
STOCK156CKC;
ALTER TABLE STOCK156 ADD CONSTRAINT
STOCK156CKC CHECK (S_W_ID BETWEEN 248001 AND
249600);
SET INTEGRITY FOR STOCK156 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK157 OFF;
ALTER TABLE STOCK157 DROP CONSTRAINT
STOCK157CKC;

```

```

ALTER TABLE STOCK157 ADD CONSTRAINT
STOCK157CKC CHECK (S_W_ID BETWEEN 249601 AND
251200);
SET INTEGRITY FOR STOCK157 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK158 OFF;
ALTER TABLE STOCK158 DROP CONSTRAINT
STOCK158CKC;
ALTER TABLE STOCK158 ADD CONSTRAINT
STOCK158CKC CHECK (S_W_ID BETWEEN 251201 AND
252800);
SET INTEGRITY FOR STOCK158 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK159 OFF;
ALTER TABLE STOCK159 DROP CONSTRAINT
STOCK159CKC;
ALTER TABLE STOCK159 ADD CONSTRAINT
STOCK159CKC CHECK (S_W_ID BETWEEN 252801 AND
254400);
SET INTEGRITY FOR STOCK159 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK160 OFF;
ALTER TABLE STOCK160 DROP CONSTRAINT
STOCK160CKC;
ALTER TABLE STOCK160 ADD CONSTRAINT
STOCK160CKC CHECK (S_W_ID >= 254401);
SET INTEGRITY FOR STOCK160 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

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 32000);
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 32001 AND
64000);
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 64001 AND
96000);
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 96001 AND
128000);
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 128001 AND
160000);
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 160001 AND
192000);
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 192001 AND
224000);
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 >= 224001);
SET INTEGRITY FOR WAREHOUSE8 ALL IMMEDIATE
UNCHECKED;
connect reset;

```

CRIDX_CUST_IDXB.ddl


```

connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB156;
CREATE INDEX CUST_IDXB156
    ON CUSTOMER156(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_IDXB157;
CREATE INDEX CUST_IDXB157
    ON CUSTOMER157(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_IDXB158;
CREATE INDEX CUST_IDXB158
    ON CUSTOMER158(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_IDXB159;
CREATE INDEX CUST_IDXB159
    ON CUSTOMER159(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_IDXB160;
CREATE INDEX CUST_IDXB160
    ON CUSTOMER160(C_LAST, C_W_ID,
C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;

```

CRIDX ORDR_IDXB.ddl

```

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB1;
CREATE INDEX ORDR_IDXB1
    ON ORDERS1(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB2;
CREATE INDEX ORDR_IDXB2
    ON ORDERS2(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB3;
CREATE INDEX ORDR_IDXB3
    ON ORDERS3(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB4;
CREATE INDEX ORDR_IDXB4
    ON ORDERS4(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB5;

```

```

CREATE INDEX ORDR_IDXB5
    ON ORDERS5(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB6;
CREATE INDEX ORDR_IDXB6
    ON ORDERS6(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB7;
CREATE INDEX ORDR_IDXB7
    ON ORDERS7(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB8;
CREATE INDEX ORDR_IDXB8
    ON ORDERS8(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB9;
CREATE INDEX ORDR_IDXB9
    ON ORDERS9(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB10;
CREATE INDEX ORDR_IDXB10
    ON ORDERS10(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB11;
CREATE INDEX ORDR_IDXB11
    ON ORDERS11(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB12;
CREATE INDEX ORDR_IDXB12
    ON ORDERS12(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB13;
CREATE INDEX ORDR_IDXB13
    ON ORDERS13(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB14;
CREATE INDEX ORDR_IDXB14
    ON ORDERS14(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB15;
CREATE INDEX ORDR_IDXB15

```

```

    ON ORDERS15(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB16;
CREATE INDEX ORDR_IDXB16
    ON ORDERS16(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB17;
CREATE INDEX ORDR_IDXB17
    ON ORDERS17(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB18;
CREATE INDEX ORDR_IDXB18
    ON ORDERS18(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB19;
CREATE INDEX ORDR_IDXB19
    ON ORDERS19(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB20;
CREATE INDEX ORDR_IDXB20
    ON ORDERS20(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB21;
CREATE INDEX ORDR_IDXB21
    ON ORDERS21(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB22;
CREATE INDEX ORDR_IDXB22
    ON ORDERS22(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB23;
CREATE INDEX ORDR_IDXB23
    ON ORDERS23(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB24;
CREATE INDEX ORDR_IDXB24
    ON ORDERS24(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB25;
CREATE INDEX ORDR_IDXB25
    ON ORDERS25(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;

```



```

connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB150;
CREATE INDEX ORDR_IDXB150
    ON ORDERS150(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB151;
CREATE INDEX ORDR_IDXB151
    ON ORDERS151(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB152;
CREATE INDEX ORDR_IDXB152
    ON ORDERS152(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB153;
CREATE INDEX ORDR_IDXB153
    ON ORDERS153(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB154;
CREATE INDEX ORDR_IDXB154
    ON ORDERS154(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB155;
CREATE INDEX ORDR_IDXB155
    ON ORDERS155(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB156;
CREATE INDEX ORDR_IDXB156
    ON ORDERS156(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB157;
CREATE INDEX ORDR_IDXB157
    ON ORDERS157(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB158;
CREATE INDEX ORDR_IDXB158
    ON ORDERS158(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB159;
CREATE INDEX ORDR_IDXB159
    ON ORDERS159(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;

```

```

DROP INDEX ORDR_IDXB160;
CREATE INDEX ORDR_IDXB160
    ON ORDERS160(O_C_ID, O_W_ID, O_D_ID,
O_ID DESC) PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

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       BIGINT       NOT NULL,
C_CREDIT_LIM  BIGINT       NOT NULL,
C_MIDDLE      CHAR(2)      NOT NULL,
C_CREDIT      CHAR(2)      NOT NULL,
C_DISCOUNT   INTEGER     NOT NULL,
C_DATA        VARCHAR(500) NOT NULL,
C_LAST        VARCHAR(16)  NOT NULL,
C_FIRST       VARCHAR(16)  NOT NULL,
C_STREET_1    VARCHAR(20)  NOT NULL,
C_STREET_2    VARCHAR(20)  NOT NULL,
C_CITY        VARCHAR(20)  NOT NULL,
C_D_ID        SMALLINT     NOT NULL,
C_W_ID        INTEGER      NOT NULL,
C_DELIVERY_CNT INTEGER     NOT NULL,
C_BALANCE     BIGINT       NOT NULL,
C_YTD_PAYMENT BIGINT       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 1600,
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       BIGINT       NOT NULL,
C_CREDIT_LIM  BIGINT       NOT NULL,
C_MIDDLE      CHAR(2)      NOT NULL,
C_CREDIT      CHAR(2)      NOT NULL,
C_DISCOUNT   INTEGER     NOT NULL,
C_DATA        VARCHAR(500) NOT NULL,
C_LAST        VARCHAR(16)  NOT NULL,
C_FIRST       VARCHAR(16)  NOT NULL,
C_STREET_1    VARCHAR(20)  NOT NULL,

```

```

C_STREET_2    VARCHAR(20)  NOT NULL,
C_CITY        VARCHAR(20)  NOT NULL,
C_D_ID        SMALLINT     NOT NULL,
C_W_ID        INTEGER      NOT NULL,
C_DELIVERY_CNT INTEGER     NOT NULL,
C_BALANCE     BIGINT       NOT NULL,
C_YTD_PAYMENT BIGINT       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 1601 ENDING AT
3200,
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       BIGINT       NOT NULL,
C_CREDIT_LIM  BIGINT       NOT NULL,
C_MIDDLE      CHAR(2)      NOT NULL,
C_CREDIT      CHAR(2)      NOT NULL,
C_DISCOUNT   INTEGER     NOT NULL,
C_DATA        VARCHAR(500) NOT NULL,
C_LAST        VARCHAR(16)  NOT NULL,
C_FIRST       VARCHAR(16)  NOT NULL,
C_STREET_1    VARCHAR(20)  NOT NULL,
C_STREET_2    VARCHAR(20)  NOT NULL,
C_CITY        VARCHAR(20)  NOT NULL,
C_D_ID        SMALLINT     NOT NULL,
C_W_ID        INTEGER      NOT NULL,
C_DELIVERY_CNT INTEGER     NOT NULL,
C_BALANCE     BIGINT       NOT NULL,
C_YTD_PAYMENT BIGINT       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 3201 ENDING AT
4800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 4801 ENDING AT
6400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 6401 ENDING AT
8000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 8001 ENDING AT
9600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,

```

```

C_YTD_PAYMENT BIGINT 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 9601 ENDING AT
11200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 11201 ENDING AT
12800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,

```



```

C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 12801 ENDING AT
14400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 14401 ENDING AT
16000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 16001 ENDING AT
17600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 17601 ENDING AT
19200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 19201 ENDING AT
20800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,

```

```

C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 20801 ENDING AT
22400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 22401 ENDING AT
24000,
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 BIGINT NOT NULL,

```

```

C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 24001 ENDING AT
25600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 25601 ENDING AT
27200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 27201 ENDING AT
28800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 28801 ENDING AT
30400,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 30401 ENDING AT
32000,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,

```

```

  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 32001 ENDING AT
33600,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 33601 ENDING AT
35200,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 35201 ENDING AT
36800,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT    NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER    NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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,

```

```

38400, C_W_ID STARTING FROM 36801 ENDING AT
)
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 38401 ENDING AT
40000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 40001 ENDING AT
41600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 41601 ENDING AT
43200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,

```

```

C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 43201 ENDING AT
44800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 44801 ENDING AT
46400,
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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 46401 ENDING AT
48000,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 48001 ENDING AT
49600,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 49601 ENDING AT
51200,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,

```

```

  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 51201 ENDING AT
52800,
  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   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   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 52801 ENDING AT
54400,
  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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 54401 ENDING AT
56000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 56001 ENDING AT
57600,

```

```

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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 57601 ENDING AT
59200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,

```

```

C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 59201 ENDING AT
60800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 60801 ENDING AT
62400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,

```

```

C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 62401 ENDING AT
64000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 64001 ENDING AT
65600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 65601 ENDING AT
67200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 67201 ENDING AT
68800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 68801 ENDING AT
70400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,

```

```

C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 70401 ENDING AT
72000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 72001 ENDING AT
73600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 73601 ENDING AT
75200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 75201 ENDING AT
76800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 76801 ENDING AT
78400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 78401 ENDING AT
80000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 80001 ENDING AT
81600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,

```

```

C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 81601 ENDING AT
83200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 83201 ENDING AT
84800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 84801 ENDING AT
86400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 86401 ENDING AT
88000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 88001 ENDING AT
89600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 89601 ENDING AT
91200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 91201 ENDING AT
92800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,

```

```

C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_059
INDEX IN is_customer_059
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 92801 ENDING AT
94400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 94401 ENDING AT
96000,
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   BIGINT      NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT      NOT NULL,
  C_YTD_PAYMENT BIGINT  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 96001 ENDING AT
97600,
  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   BIGINT      NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT      NOT NULL,
  C_YTD_PAYMENT BIGINT  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 97601 ENDING AT
99200,
  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   BIGINT      NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT      NOT NULL,
  C_YTD_PAYMENT BIGINT  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 99201 ENDING AT
100800,
  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   BIGINT      NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,

```

```

  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT      NOT NULL,
  C_YTD_PAYMENT BIGINT  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 100801 ENDING AT
102400,
  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   BIGINT      NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT    NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT      NOT NULL,
  C_YTD_PAYMENT BIGINT  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 102401 ENDING AT
104000,
  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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 104001 ENDING AT
105600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 105601 ENDING AT
107200,

```

```

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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 107201 ENDING AT
108800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,

```

```

C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 108801 ENDING AT
110400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 110401 ENDING AT
112000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,

```

```

C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 112001 ENDING AT
113600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 113601 ENDING AT
115200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 115201 ENDING AT
116800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 116801 ENDING AT
118400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 118401 ENDING AT
120000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,

```

```

C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 120001 ENDING AT
121600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 121601 ENDING AT
123200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 123201 ENDING AT
124800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 124801 ENDING AT
126400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 126401 ENDING AT
128000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 128001 ENDING AT
129600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 129601 ENDING AT
131200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,

```

```

C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 131201 ENDING AT
132800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 132801 ENDING AT
134400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 134401 ENDING AT
136000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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,

```

```

137600, C_W_ID STARTING FROM 136001 ENDING AT
)
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 137601 ENDING AT
139200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 139201 ENDING AT
140800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 140801 ENDING AT
142400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,

```

```

C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 142401 ENDING AT
144000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 144001 ENDING AT
145600,
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   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT 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 145601 ENDING AT
147200,
  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   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT 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 147201 ENDING AT
148800,
  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   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT 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 148801 ENDING AT
150400,
  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   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT NOT NULL,
  C_PAYMENT_CNT INTEGER NOT NULL
)

```

```

C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY    VARCHAR(20) NOT NULL,
C_D_ID    SMALLINT   NOT NULL,
C_W_ID    INTEGER     NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT 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 150401 ENDING AT
152000,
  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   BIGINT     NOT NULL,
  C_CREDIT_LIM BIGINT   NOT NULL,
  C_MIDDLE  CHAR(2)     NOT NULL,
  C_CREDIT  CHAR(2)     NOT NULL,
  C_DISCOUNT INTEGER   NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16) NOT NULL,
  C_FIRST   VARCHAR(16) NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20) NOT NULL,
  C_D_ID    SMALLINT   NOT NULL,
  C_W_ID    INTEGER     NOT NULL,
  C_DELIVERY_CNT INTEGER NOT NULL,
  C_BALANCE BIGINT     NOT NULL,
  C_YTD_PAYMENT BIGINT 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 152001 ENDING AT
153600,
  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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 153601 ENDING AT
155200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 155201 ENDING AT
156800,

```

```

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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 156801 ENDING AT
158400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,

```

```

C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 158401 ENDING AT
160000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 160001 ENDING AT
161600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,

```

```

C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 161601 ENDING AT
163200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 163201 ENDING AT
164800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 164801 ENDING AT
166400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 166401 ENDING AT
168000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 168001 ENDING AT
169600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,

```

```

C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 169601 ENDING AT
171200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 171201 ENDING AT
172800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 172801 ENDING AT
174400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 174401 ENDING AT
176000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 176001 ENDING AT
177600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 177601 ENDING AT
179200,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 179201 ENDING AT
180800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,

```

```

C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 180801 ENDING AT
182400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 182401 ENDING AT
184000,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 184001 ENDING AT
185600,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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,

```

```

187200, C_W_ID STARTING FROM 185601 ENDING AT
)
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 187201 ENDING AT
188800,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 188801 ENDING AT
190400,
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 BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT 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 190401 ENDING AT
192000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER121;
CREATE TABLE CUSTOMER121
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,

```

```

C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_121
INDEX IN is_customer_121
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 192001 ENDING AT
193600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER122;
CREATE TABLE CUSTOMER122
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_122
INDEX IN is_customer_122
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 193601 ENDING AT
195200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER123;
CREATE TABLE CUSTOMER123
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_123
INDEX IN is_customer_123
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 195201 ENDING AT
196800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER124;
CREATE TABLE CUSTOMER124
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)

```

```

IN ts_customer_124
INDEX IN is_customer_124
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 196801 ENDING AT
198400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER125;
CREATE TABLE CUSTOMER125
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_125
INDEX IN is_customer_125
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 198401 ENDING AT
200000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER126;
CREATE TABLE CUSTOMER126
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)

```

```

C_STREET_1  VARCHAR(20) NOT NULL,
C_STREET_2  VARCHAR(20) NOT NULL,
C_CITY      VARCHAR(20) NOT NULL,
C_D_ID      SMALLINT   NOT NULL,
C_W_ID      INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE   BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT   NOT NULL,
C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_126
INDEX IN is_customer_126
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 200001 ENDING AT
201600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER127;
CREATE TABLE CUSTOMER127
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_127
INDEX IN is_customer_127
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 201601 ENDING AT
203200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER128;
CREATE TABLE CUSTOMER128
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,

```

```

C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_128
INDEX IN is_customer_128
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 203201 ENDING AT
204800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER129;
CREATE TABLE CUSTOMER129
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_129
INDEX IN is_customer_129
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 204801 ENDING AT
206400,

```

```

C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER130;
CREATE TABLE CUSTOMER130
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_130
INDEX IN is_customer_130
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 206401 ENDING AT
208000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER131;
CREATE TABLE CUSTOMER131
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,

```

```

C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_131
INDEX IN is_customer_131
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 208001 ENDING AT
209600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER132;
CREATE TABLE CUSTOMER132
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_132
INDEX IN is_customer_132
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 209601 ENDING AT
211200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER133;
CREATE TABLE CUSTOMER133
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,

```



```

C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_133
INDEX IN is_customer_133
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 211201 ENDING AT
212800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER134;
CREATE TABLE CUSTOMER134
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_134
INDEX IN is_customer_134
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 212801 ENDING AT
214400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER135;

```

```

CREATE TABLE CUSTOMER135
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_135
INDEX IN is_customer_135
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 214401 ENDING AT
216000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER136;
CREATE TABLE CUSTOMER136
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_136
INDEX IN is_customer_136

```

```

ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 216001 ENDING AT
217600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER137;
CREATE TABLE CUSTOMER137
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_137
INDEX IN is_customer_137
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 217601 ENDING AT
219200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER138;
CREATE TABLE CUSTOMER138
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,

```

```

C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_138
INDEX IN is_customer_138
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 219201 ENDING AT
220800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER139;
CREATE TABLE CUSTOMER139
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_139
INDEX IN is_customer_139
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 220801 ENDING AT
222400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER140;
CREATE TABLE CUSTOMER140
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,

```

```

C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_140
INDEX IN is_customer_140
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 222401 ENDING AT
224000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER141;
CREATE TABLE CUSTOMER141
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_141
INDEX IN is_customer_141
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 224001 ENDING AT
225600,
C_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER142;
CREATE TABLE CUSTOMER142
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_142
INDEX IN is_customer_142
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 225601 ENDING AT
227200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER143;
CREATE TABLE CUSTOMER143
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,

```

```

C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_143
INDEX IN is_customer_143
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 227201 ENDING AT
228800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER144;
CREATE TABLE CUSTOMER144
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_144
INDEX IN is_customer_144
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 228801 ENDING AT
230400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER145;
CREATE TABLE CUSTOMER145
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,

```

```

C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_145
INDEX IN is_customer_145
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 230401 ENDING AT
232000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER146;
CREATE TABLE CUSTOMER146
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_146
INDEX IN is_customer_146
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 232001 ENDING AT
233600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER147;
CREATE TABLE CUSTOMER147
(

```

```

C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_147
INDEX IN is_customer_147
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 233601 ENDING AT
235200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER148;
CREATE TABLE CUSTOMER148
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_148
INDEX IN is_customer_148
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 235201 ENDING AT
236800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER149;
CREATE TABLE CUSTOMER149
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_149
INDEX IN is_customer_149
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 236801 ENDING AT
238400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER150;
CREATE TABLE CUSTOMER150
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_150
INDEX IN is_customer_150
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 238401 ENDING AT
240000,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER151;
CREATE TABLE CUSTOMER151
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_151
INDEX IN is_customer_151
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 240001 ENDING AT
241600,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER152;
CREATE TABLE CUSTOMER152
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,

```

```

C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_152
INDEX IN is_customer_152
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 241601 ENDING AT
243200,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER153;
CREATE TABLE CUSTOMER153
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_153
INDEX IN is_customer_153
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 243201 ENDING AT
244800,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE CUSTOMER154;
CREATE TABLE CUSTOMER154
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_154
INDEX IN is_customer_154
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 244801 ENDING AT
246400,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER155;
CREATE TABLE CUSTOMER155
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)

```

```

IN ts_customer_155
INDEX IN is_customer_155
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 246401 ENDING AT
248000,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER156;
CREATE TABLE CUSTOMER156
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_156
INDEX IN is_customer_156
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 248001 ENDING AT
249600,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER157;
CREATE TABLE CUSTOMER157
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)

```

```

C_STREET_1  VARCHAR(20) NOT NULL,
C_STREET_2  VARCHAR(20) NOT NULL,
C_CITY      VARCHAR(20) NOT NULL,
C_D_ID      SMALLINT   NOT NULL,
C_W_ID      INTEGER    NOT NULL,
C_DELIVERY_CNT INTEGER  NOT NULL,
C_BALANCE   BIGINT     NOT NULL,
C_YTD_PAYMENT BIGINT   NOT NULL,
C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_157
INDEX IN is_customer_157
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 249601 ENDING AT
251200,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER158;
CREATE TABLE CUSTOMER158
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,
  C_ZIP     CHAR(9)      NOT NULL,
  C_PHONE   CHAR(16)     NOT NULL,
  C_SINCE   BIGINT       NOT NULL,
  C_CREDIT_LIM BIGINT     NOT NULL,
  C_MIDDLE  CHAR(2)      NOT NULL,
  C_CREDIT  CHAR(2)      NOT NULL,
  C_DISCOUNT INTEGER     NOT NULL,
  C_DATA    VARCHAR(500) NOT NULL,
  C_LAST    VARCHAR(16)  NOT NULL,
  C_FIRST   VARCHAR(16)  NOT NULL,
  C_STREET_1 VARCHAR(20) NOT NULL,
  C_STREET_2 VARCHAR(20) NOT NULL,
  C_CITY    VARCHAR(20)  NOT NULL,
  C_D_ID    SMALLINT     NOT NULL,
  C_W_ID    INTEGER      NOT NULL,
  C_DELIVERY_CNT INTEGER  NOT NULL,
  C_BALANCE BIGINT       NOT NULL,
  C_YTD_PAYMENT BIGINT   NOT NULL,
  C_PAYMENT_CNT INTEGER   NOT NULL
)
IN ts_customer_158
INDEX IN is_customer_158
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 251201 ENDING AT
252800,
  C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER159;
CREATE TABLE CUSTOMER159
(
  C_ID      INTEGER      NOT NULL,
  C_STATE   CHAR(2)      NOT NULL,

```

```

C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_159
INDEX IN is_customer_159
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 252801 ENDING AT
254400,
C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER160;
CREATE TABLE CUSTOMER160
(
C_ID INTEGER NOT NULL,
C_STATE CHAR(2) NOT NULL,
C_ZIP CHAR(9) NOT NULL,
C_PHONE CHAR(16) NOT NULL,
C_SINCE BIGINT NOT NULL,
C_CREDIT_LIM BIGINT NOT NULL,
C_MIDDLE CHAR(2) NOT NULL,
C_CREDIT CHAR(2) NOT NULL,
C_DISCOUNT INTEGER NOT NULL,
C_DATA VARCHAR(500) NOT NULL,
C_LAST VARCHAR(16) NOT NULL,
C_FIRST VARCHAR(16) NOT NULL,
C_STREET_1 VARCHAR(20) NOT NULL,
C_STREET_2 VARCHAR(20) NOT NULL,
C_CITY VARCHAR(20) NOT NULL,
C_D_ID SMALLINT NOT NULL,
C_W_ID INTEGER NOT NULL,
C_DELIVERY_CNT INTEGER NOT NULL,
C_BALANCE BIGINT NOT NULL,
C_YTD_PAYMENT BIGINT NOT NULL,
C_PAYMENT_CNT INTEGER NOT NULL
)
IN ts_customer_160
INDEX IN is_customer_160
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 254401 ENDING AT
256000,

```

```

C_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

CRTB DISTRICT.ddl

connect to TPCC in share mode;
DROP TABLE DISTRICT1;
CREATE TABLE DISTRICT1
(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_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 32000
)
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 INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_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 32001 ENDING AT
64000
)
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 INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_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 64001 ENDING AT
96000
)
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 INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_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 96001 ENDING AT
128000
)
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 INTEGER NOT NULL,
D_YTD BIGINT NOT NULL,
D_NAME CHAR(10) NOT NULL,
D_STREET_1 CHAR(20) NOT NULL,
D_STREET_2 CHAR(20) NOT NULL,
D_CITY CHAR(20) NOT NULL,
D_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 128001 ENDING AT
160000
)
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 INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_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 160001 ENDING AT
192000
)
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 INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_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 192001 ENDING AT
224000
)
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 INTEGER NOT NULL,
  D_YTD BIGINT NOT NULL,
  D_NAME CHAR(10) NOT NULL,
  D_STREET_1 CHAR(20) NOT NULL,
  D_STREET_2 CHAR(20) NOT NULL,
  D_CITY CHAR(20) NOT NULL,
  D_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 224001 ENDING AT
256000
)
ALLOW OVERFLOW;
connect reset;

```

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 BIGINT NOT NULL,
  H_AMOUNT INTEGER 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 BIGINT NOT NULL,
  H_AMOUNT INTEGER 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 BIGINT NOT NULL,
  H_AMOUNT INTEGER 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 BIGINT NOT NULL,
  H_AMOUNT INTEGER 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 BIGINT NOT NULL,
  H_AMOUNT INTEGER 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 BIGINT NOT NULL,
  H_AMOUNT INTEGER 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      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER 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      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_008
INDEX IN ts_history_008;
ALTER TABLE HISTORY8 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY9;
CREATE TABLE HISTORY9
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_009
INDEX IN ts_history_009;
ALTER TABLE HISTORY9 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY10;
CREATE TABLE HISTORY10
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,

```

```

    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_010
INDEX IN ts_history_010;
ALTER TABLE HISTORY10 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY11;
CREATE TABLE HISTORY11
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_011
INDEX IN ts_history_011;
ALTER TABLE HISTORY11 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY12;
CREATE TABLE HISTORY12
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_012
INDEX IN ts_history_012;
ALTER TABLE HISTORY12 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY13;
CREATE TABLE HISTORY13
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_013
INDEX IN ts_history_013;
ALTER TABLE HISTORY13 APPEND ON;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE HISTORY14;
CREATE TABLE HISTORY14
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_014
INDEX IN ts_history_014;
ALTER TABLE HISTORY14 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY15;
CREATE TABLE HISTORY15
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_015
INDEX IN ts_history_015;
ALTER TABLE HISTORY15 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY16;
CREATE TABLE HISTORY16
(
    H_C_ID      INTEGER NOT NULL,
    H_C_D_ID    SMALLINT NOT NULL,
    H_C_W_ID    INTEGER NOT NULL,
    H_D_ID      SMALLINT NOT NULL,
    H_W_ID      INTEGER NOT NULL,
    H_DATE      BIGINT  NOT NULL,
    H_AMOUNT    INTEGER NOT NULL,
    H_DATA      CHAR(24) NOT NULL
)
IN ts_history_016
INDEX IN ts_history_016;
ALTER TABLE HISTORY16 APPEND ON;
connect reset;

CRTB_ITEM.ddl

connect to TPCC in share mode;
DROP TABLE ITEM;
CREATE TABLE ITEM
(
    I_NAME      CHAR(24) NOT NULL,
    I_PRICE     INTEGER 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 100000
)
ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;
connect reset;

```

CRTB NEW ORDER.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
32000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
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 32001 ENDING AT
64000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
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 64001 ENDING AT
96000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
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 96001 ENDING AT
128000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
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 128001 ENDING AT
160000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
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 160001 ENDING AT
192000,

```

```

NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
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 192001 ENDING AT
224000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
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 224001 ENDING AT
256000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 1900 ENDING AT
3694
)
ALLOW OVERFLOW;
connect reset;
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
32000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
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 32001 ENDING AT
64000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
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 64001 ENDING AT
96000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
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 96001 ENDING AT
128000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
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 128001 ENDING AT
160000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
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 160001 ENDING AT
192000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
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 192001 ENDING AT
224000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
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 224001 ENDING AT
256000,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3695 ENDING AT
5489
)
ALLOW OVERFLOW;
connect reset;

```

CRTB ORDERS.ddl

```

connect to TPCC in share mode;
DROP TABLE ORDERS1;
CREATE TABLE ORDERS1
(
  O_C_ID    INTEGER    NOT NULL,
  O_ENTRY_D BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT   NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 1 ENDING AT 1600,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT   NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 1601 ENDING AT
3200,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 3201 ENDING AT
4800,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 4801 ENDING AT
6400,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,

```

```

  O_W_ID    INTEGER    NOT NULL,
  O_D_ID    SMALLINT   NOT NULL
)
O_W_ID STARTING FROM 6401 ENDING AT
8000,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 8001 ENDING AT
9600,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 9601 ENDING AT
11200,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 11201 ENDING AT
12800,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 12801 ENDING AT
14400,
  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 BIGINT     NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT  NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 14401 ENDING AT
16000,
  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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 16001 ENDING AT
17600,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 17601 ENDING AT
19200,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 19201 ENDING AT
20800,

```

```

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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 20801 ENDING AT
22400,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 22401 ENDING AT
24000,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 24001 ENDING AT
25600,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 25601 ENDING AT
27200,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 27201 ENDING AT
28800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 28801 ENDING AT
30400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 30401 ENDING AT
32000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 32001 ENDING AT
33600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 33601 ENDING AT
35200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 35201 ENDING AT
36800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 36801 ENDING AT
38400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 38401 ENDING AT
40000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 40001 ENDING AT
41600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,

```

```

O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 41601 ENDING AT
43200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 43201 ENDING AT
44800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 44801 ENDING AT
46400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 46401 ENDING AT
48000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 48001 ENDING AT
49600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 49601 ENDING AT
51200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 51201 ENDING AT
52800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 52801 ENDING AT
54400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 54401 ENDING AT
56000,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 56001 ENDING AT
57600,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 57601 ENDING AT
59200,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 59201 ENDING AT
60800,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 60801 ENDING AT
62400,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,

```

```

O_W_ID STARTING FROM 62401 ENDING AT
64000,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 64001 ENDING AT
65600,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 65601 ENDING AT
67200,
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 BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 67201 ENDING AT
68800,
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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 68801 ENDING AT
70400,
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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 70401 ENDING AT
72000,
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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 72001 ENDING AT
73600,
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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 73601 ENDING AT
75200,
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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 75201 ENDING AT
76800,

```

```

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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 76801 ENDING AT
78400,
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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 78401 ENDING AT
80000,
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 BIGINT     NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 80001 ENDING AT
81600,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 81601 ENDING AT
83200,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 83201 ENDING AT
84800,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 84801 ENDING AT
86400,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 86401 ENDING AT
88000,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 88001 ENDING AT
89600,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 89601 ENDING AT
91200,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 91201 ENDING AT
92800,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT   SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 92801 ENDING AT
94400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 94401 ENDING AT
96000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 96001 ENDING AT
97600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,

```

```

O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 97601 ENDING AT
99200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 99201 ENDING AT
100800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 100801 ENDING AT
102400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 102401 ENDING AT
104000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 104001 ENDING AT
105600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 105601 ENDING AT
107200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 107201 ENDING AT
108800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 108801 ENDING AT
110400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 110401 ENDING AT
112000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 112001 ENDING AT
113600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 113601 ENDING AT
115200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 115201 ENDING AT
116800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 116801 ENDING AT
118400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,

```

```

O_W_ID STARTING FROM 118401 ENDING AT
120000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 120001 ENDING AT
121600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 121601 ENDING AT
123200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 123201 ENDING AT
124800,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 124801 ENDING AT
126400,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 126401 ENDING AT
128000,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 128001 ENDING AT
129600,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 129601 ENDING AT
131200,
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 BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 131201 ENDING AT
132800,

```

```

        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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_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 0 ENDING AT 3694,
    O_W_ID STARTING FROM 132801 ENDING AT
134400,
    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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_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 0 ENDING AT 3694,
    O_W_ID STARTING FROM 134401 ENDING AT
136000,
    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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_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 0 ENDING AT 3694,
    O_W_ID STARTING FROM 136001 ENDING AT
137600,
    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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_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 0 ENDING AT 3694,
    O_W_ID STARTING FROM 137601 ENDING AT
139200,
    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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_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 0 ENDING AT 3694,
    O_W_ID STARTING FROM 139201 ENDING AT
140800,
    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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_LOCAL SMALLINT NOT NULL,
    O_ID      INTEGER    NOT NULL,
    O_W_ID    INTEGER    NOT NULL,
    O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_089
INDEX IN is_order_089
ORGANIZE BY KEY SEQUENCE (
    O_ID STARTING FROM 0 ENDING AT 3694,
    O_W_ID STARTING FROM 140801 ENDING AT
142400,
    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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_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 0 ENDING AT 3694,
    O_W_ID STARTING FROM 142401 ENDING AT
144000,
    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 BIGINT    NOT NULL,
    O_CARRIER_ID SMALLINT NOT NULL,
    O_OL_CNT  SMALLINT  NOT NULL,
    O_ALL_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 0 ENDING AT 3694,
    O_W_ID STARTING FROM 144001 ENDING AT
145600,
    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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 145601 ENDING AT
147200,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 147201 ENDING AT
148800,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 148801 ENDING AT
150400,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 150401 ENDING AT
152000,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 152001 ENDING AT
153600,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,

```

```

  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 153601 ENDING AT
155200,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 155201 ENDING AT
156800,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 156801 ENDING AT
158400,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 158401 ENDING AT
160000,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 160001 ENDING AT
161600,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 161601 ENDING AT
163200,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 163201 ENDING AT
164800,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 164801 ENDING AT
166400,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 166401 ENDING AT
168000,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 168001 ENDING AT
169600,
  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 BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 169601 ENDING AT
171200,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 171201 ENDING AT
172800,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 172801 ENDING AT
174400,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,

```

```

  O_W_ID STARTING FROM 174401 ENDING AT
176000,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 176001 ENDING AT
177600,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 177601 ENDING AT
179200,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_LOCAL 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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 179201 ENDING AT
180800,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 180801 ENDING AT
182400,
  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   BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT    SMALLINT NOT NULL,
  O_ALL_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 0 ENDING AT 3694,
  O_W_ID STARTING FROM 182401 ENDING AT
184000,
  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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 184001 ENDING AT
185600,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 185601 ENDING AT
187200,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 187201 ENDING AT
188800,

```

```

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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 188801 ENDING AT
190400,
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 BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_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 0 ENDING AT 3694,
O_W_ID STARTING FROM 190401 ENDING AT
192000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS121;
CREATE TABLE ORDERS121
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)

```

```

)
IN ts_order_121
INDEX IN is_order_121
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 192001 ENDING AT
193600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS122;
CREATE TABLE ORDERS122
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_122
INDEX IN is_order_122
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 193601 ENDING AT
195200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS123;
CREATE TABLE ORDERS123
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_123
INDEX IN is_order_123
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 195201 ENDING AT
196800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS124;
CREATE TABLE ORDERS124
(
O_C_ID    INTEGER    NOT NULL,

```

```

O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_124
INDEX IN is_order_124
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 196801 ENDING AT
198400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS125;
CREATE TABLE ORDERS125
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_125
INDEX IN is_order_125
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 198401 ENDING AT
200000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS126;
CREATE TABLE ORDERS126
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_126
INDEX IN is_order_126
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 200001 ENDING AT
201600,
O_D_ID STARTING FROM 1 ENDING AT 10
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS127;
CREATE TABLE ORDERS127
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_127
INDEX IN is_order_127
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 201601 ENDING AT
203200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS128;
CREATE TABLE ORDERS128
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_128
INDEX IN is_order_128
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 203201 ENDING AT
204800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS129;
CREATE TABLE ORDERS129
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_129

```

```

INDEX IN is_order_129
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 204801 ENDING AT
206400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS130;
CREATE TABLE ORDERS130
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_130
INDEX IN is_order_130
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 206401 ENDING AT
208000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS131;
CREATE TABLE ORDERS131
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_131
INDEX IN is_order_131
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 208001 ENDING AT
209600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS132;
CREATE TABLE ORDERS132
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,

```

```

O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_132
INDEX IN is_order_132
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 209601 ENDING AT
211200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS133;
CREATE TABLE ORDERS133
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_133
INDEX IN is_order_133
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 211201 ENDING AT
212800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS134;
CREATE TABLE ORDERS134
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_134
INDEX IN is_order_134
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 212801 ENDING AT
214400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE ORDERS135;
CREATE TABLE ORDERS135
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_135
INDEX IN is_order_135
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 214401 ENDING AT
216000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS136;
CREATE TABLE ORDERS136
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_136
INDEX IN is_order_136
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 216001 ENDING AT
217600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS137;
CREATE TABLE ORDERS137
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_137
INDEX IN is_order_137
ORGANIZE BY KEY SEQUENCE (

```

```

O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 217601 ENDING AT
219200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS138;
CREATE TABLE ORDERS138
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_138
INDEX IN is_order_138
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 219201 ENDING AT
220800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS139;
CREATE TABLE ORDERS139
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID INTEGER NOT NULL,
O_W_ID INTEGER NOT NULL,
O_D_ID SMALLINT NOT NULL
)
IN ts_order_139
INDEX IN is_order_139
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 220801 ENDING AT
222400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS140;
CREATE TABLE ORDERS140
(
O_C_ID INTEGER NOT NULL,
O_ENTRY_D BIGINT NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,

```

```

O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_140
INDEX IN is_order_140
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 222401 ENDING AT
224000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS141;
CREATE TABLE ORDERS141
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_141
INDEX IN is_order_141
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 224001 ENDING AT
225600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS142;
CREATE TABLE ORDERS142
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_142
INDEX IN is_order_142
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 225601 ENDING AT
227200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS143;

```

```

CREATE TABLE ORDERS143
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_143
INDEX IN is_order_143
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 227201 ENDING AT
228800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS144;
CREATE TABLE ORDERS144
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_144
INDEX IN is_order_144
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 228801 ENDING AT
230400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS145;
CREATE TABLE ORDERS145
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_145
INDEX IN is_order_145
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,

```

```

O_W_ID STARTING FROM 230401 ENDING AT
232000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS146;
CREATE TABLE ORDERS146
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_146
INDEX IN is_order_146
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 232001 ENDING AT
233600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS147;
CREATE TABLE ORDERS147
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,
O_W_ID    INTEGER NOT NULL,
O_D_ID    SMALLINT NOT NULL
)
IN ts_order_147
INDEX IN is_order_147
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 233601 ENDING AT
235200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS148;
CREATE TABLE ORDERS148
(
O_C_ID    INTEGER NOT NULL,
O_ENTRY_D BIGINT  NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER NOT NULL,

```

```

O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_148
INDEX IN is_order_148
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 235201 ENDING AT
236800,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS149;
CREATE TABLE ORDERS149
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_149
INDEX IN is_order_149
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 236801 ENDING AT
238400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS150;
CREATE TABLE ORDERS150
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_150
INDEX IN is_order_150
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 238401 ENDING AT
240000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS151;
CREATE TABLE ORDERS151

```

```

(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_151
INDEX IN is_order_151
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 240001 ENDING AT
241600,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS152;
CREATE TABLE ORDERS152
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_152
INDEX IN is_order_152
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 241601 ENDING AT
243200,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS153;
CREATE TABLE ORDERS153
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_153
INDEX IN is_order_153
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 243201 ENDING AT
244800,

```

```

O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS154;
CREATE TABLE ORDERS154
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_154
INDEX IN is_order_154
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 244801 ENDING AT
246400,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS155;
CREATE TABLE ORDERS155
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL
)
IN ts_order_155
INDEX IN is_order_155
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 0 ENDING AT 3694,
O_W_ID STARTING FROM 246401 ENDING AT
248000,
O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS156;
CREATE TABLE ORDERS156
(
O_C_ID    INTEGER    NOT NULL,
O_ENTRY_D BIGINT    NOT NULL,
O_CARRIER_ID SMALLINT NOT NULL,
O_OL_CNT  SMALLINT  NOT NULL,
O_ALL_LOCAL SMALLINT NOT NULL,
O_ID      INTEGER    NOT NULL,
O_W_ID    INTEGER    NOT NULL,
O_D_ID    SMALLINT  NOT NULL

```

```

)
IN ts_order_156
INDEX IN is_order_156
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3694,
  O_W_ID STARTING FROM 248001 ENDING AT
249600,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS157;
CREATE TABLE ORDERS157
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER NOT NULL,
  O_W_ID    INTEGER NOT NULL,
  O_D_ID    SMALLINT NOT NULL
)
IN ts_order_157
INDEX IN is_order_157
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3694,
  O_W_ID STARTING FROM 249601 ENDING AT
251200,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS158;
CREATE TABLE ORDERS158
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER NOT NULL,
  O_W_ID    INTEGER NOT NULL,
  O_D_ID    SMALLINT NOT NULL
)
IN ts_order_158
INDEX IN is_order_158
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3694,
  O_W_ID STARTING FROM 251201 ENDING AT
252800,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS159;
CREATE TABLE ORDERS159
(
  O_C_ID    INTEGER NOT NULL,

```

```

  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER NOT NULL,
  O_W_ID    INTEGER NOT NULL,
  O_D_ID    SMALLINT NOT NULL
)
IN ts_order_159
INDEX IN is_order_159
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3694,
  O_W_ID STARTING FROM 252801 ENDING AT
254400,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS160;
CREATE TABLE ORDERS160
(
  O_C_ID    INTEGER NOT NULL,
  O_ENTRY_D BIGINT  NOT NULL,
  O_CARRIER_ID SMALLINT NOT NULL,
  O_OL_CNT  SMALLINT NOT NULL,
  O_ALL_LOCAL SMALLINT NOT NULL,
  O_ID      INTEGER NOT NULL,
  O_W_ID    INTEGER NOT NULL,
  O_D_ID    SMALLINT NOT NULL
)
IN ts_order_160
INDEX IN is_order_160
ORGANIZE BY KEY SEQUENCE (
  O_ID STARTING FROM 0 ENDING AT 3694,
  O_W_ID STARTING FROM 254401 ENDING AT
256000,
  O_D_ID STARTING FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;

CRTB ORDER LINE.ddl
connect to TPCC in share mode;
DROP TABLE ORDER_LINE1;
CREATE TABLE ORDER_LINE1
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 1600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 1601 ENDING AT
3200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 3201 ENDING AT
4800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE4;

```

<pre> CREATE TABLE ORDER_LINE4 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 4801 ENDING AT 6400, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 6401 ENDING AT 8000, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, </pre>	<pre> 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 8001 ENDING AT 9600, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 9601 ENDING AT 11200, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 11201 ENDING AT 12800, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, </pre>	<pre> 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 12801 ENDING AT 14400, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, OL_W_ID INTEGER NOT NULL, OL_NUMBER SMALLINT NOT NULL) IN ts_orderline_010 INDEX IN ts_orderline_010 ORGANIZE BY KEY SEQUENCE (OL_W_ID STARTING FROM 14401 ENDING AT 16000, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, </pre>
---	---	---

```

OL_I_ID    INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 16001 ENDING AT
17600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 17601 ENDING AT
19200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 19201 ENDING AT
20800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 20801 ENDING AT
22400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 22401 ENDING AT
24000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 24001 ENDING AT
25600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 25601 ENDING AT
27200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,

```



```

OL_O_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 27201 ENDING AT
28800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 28801 ENDING AT
30400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 30401 ENDING AT
32000,

```

```

  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 32001 ENDING AT
33600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 33601 ENDING AT
35200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 35201 ENDING AT
36800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 36801 ENDING AT
38400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 38401 ENDING AT
40000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 40001 ENDING AT
41600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 41601 ENDING AT
43200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 43201 ENDING AT
44800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 44801 ENDING AT
46400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 46401 ENDING AT
48000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 48001 ENDING AT
49600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 (

```

```

51200, OL_W_ID STARTING FROM 49601 ENDING AT
      OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,
      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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 51201 ENDING AT
52800, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,
      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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 52801 ENDING AT
54400, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,
      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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 54401 ENDING AT
56000, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,
      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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 56001 ENDING AT
57600, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,
      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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 57601 ENDING AT
59200, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,
      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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 59201 ENDING AT
60800, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,
      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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 60801 ENDING AT
62400, OL_D_ID STARTING FROM 1 ENDING AT 10,
      OL_O_ID STARTING FROM 0 ENDING AT 3694,

```

```

        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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 62401 ENDING AT
64000,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 64001 ENDING AT
65600,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,

```

```

    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 65601 ENDING AT
67200,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 67201 ENDING AT
68800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 68801 ENDING AT
70400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 70401 ENDING AT
72000,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 72001 ENDING AT
73600,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

```



```

OL_DELIVERY_D BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 (
86400, OL_W_ID STARTING FROM 84801 ENDING AT
)
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
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 BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 (
88000, OL_W_ID STARTING FROM 86401 ENDING AT
)
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
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 BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 (
89600, OL_W_ID STARTING FROM 88001 ENDING AT
)
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
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 BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 (
91200, OL_W_ID STARTING FROM 89601 ENDING AT
)
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
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 BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 (
92800, OL_W_ID STARTING FROM 91201 ENDING AT
)
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
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 BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 (
94400, OL_W_ID STARTING FROM 92801 ENDING AT
)
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
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 BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 (
96000, OL_W_ID STARTING FROM 94401 ENDING AT
)
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
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 BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 96001 ENDING AT
97600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 97601 ENDING AT
99200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 99201 ENDING AT
100800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 100801 ENDING AT
102400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 102401 ENDING AT
104000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 104001 ENDING AT
105600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 105601 ENDING AT
107200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 107201 ENDING AT
108800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 108801 ENDING AT
110400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 110401 ENDING AT
112000,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,

```

```

    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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 112001 ENDING AT
113600,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 113601 ENDING AT
115200,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,

```

```

    OL_I_ID    INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 115201 ENDING AT
116800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 116801 ENDING AT
118400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 118401 ENDING AT
120000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 120001 ENDING AT
121600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 121601 ENDING AT
123200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 123201 ENDING AT
124800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 124801 ENDING AT
126400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,

```

```

  OL_O_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 126401 ENDING AT
128000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 128001 ENDING AT
129600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 129601 ENDING AT
131200,

```

```

        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 0 ENDING AT 3694,
        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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 131201 ENDING AT
132800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 132801 ENDING AT
134400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
        OL_AMOUNT INTEGER NOT NULL,
        OL_I_ID INTEGER NOT NULL,
        OL_SUPPLY_W_ID INTEGER NOT NULL,
        OL_QUANTITY SMALLINT NOT NULL,
        OL_DIST_INFO CHAR(24) NOT NULL,
        OL_O_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 134401 ENDING AT
136000,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 0 ENDING AT 3694,
        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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 136001 ENDING AT
137600,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 137601 ENDING AT
139200,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 0 ENDING AT 3694,
        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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 139201 ENDING AT
140800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_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 140801 ENDING AT
142400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 142401 ENDING AT
144000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 144001 ENDING AT
145600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 145601 ENDING AT
147200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 147201 ENDING AT
148800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 148801 ENDING AT
150400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 150401 ENDING AT
152000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 152001 ENDING AT
153600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE97;

```

<pre> CREATE TABLE ORDER_LINE97 (OL_DELIVERY_D BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 153601 ENDING AT 155200, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 155201 ENDING AT 156800, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_ID INTEGER NOT NULL, OL_D_ID SMALLINT NOT NULL, </pre>	<pre> 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 156801 ENDING AT 158400, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 158401 ENDING AT 160000, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 160001 ENDING AT 161600, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, </pre>	<pre> 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 161601 ENDING AT 163200, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, OL_I_ID INTEGER NOT NULL, OL_SUPPLY_W_ID INTEGER NOT NULL, OL_QUANTITY SMALLINT NOT NULL, OL_DIST_INFO CHAR(24) NOT NULL, OL_O_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 163201 ENDING AT 164800, OL_D_ID STARTING FROM 1 ENDING AT 10, OL_O_ID STARTING FROM 0 ENDING AT 3694, 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 BIGINT NOT NULL, OL_AMOUNT INTEGER NOT NULL, </pre>
--	--	---

```

OL_I_ID    INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_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 164801 ENDING AT
166400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 166401 ENDING AT
168000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 168001 ENDING AT
169600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 169601 ENDING AT
171200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 171201 ENDING AT
172800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 172801 ENDING AT
174400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 174401 ENDING AT
176000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY  SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,

```

```

OL_O_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 176001 ENDING AT
177600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 177601 ENDING AT
179200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 179201 ENDING AT
180800,

```

```

  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 180801 ENDING AT
182400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 182401 ENDING AT
184000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 184001 ENDING AT
185600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 185601 ENDING AT
187200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT     INTEGER NOT NULL,
  OL_I_ID       INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY   SMALLINT NOT NULL,
  OL_DIST_INFO  CHAR(24) NOT NULL,
  OL_O_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 187201 ENDING AT
188800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 188801 ENDING AT
190400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  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 BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_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 190401 ENDING AT
192000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE121;
CREATE TABLE ORDER_LINE121
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_121
INDEX IN ts_orderline_121
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 192001 ENDING AT
193600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE122;
CREATE TABLE ORDER_LINE122
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_122
INDEX IN ts_orderline_122
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 193601 ENDING AT
195200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE123;
CREATE TABLE ORDER_LINE123
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_123
INDEX IN ts_orderline_123
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 195201 ENDING AT
196800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE124;
CREATE TABLE ORDER_LINE124
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_124
INDEX IN ts_orderline_124
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 196801 ENDING AT
198400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE125;
CREATE TABLE ORDER_LINE125
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_125
INDEX IN ts_orderline_125
ORGANIZE BY KEY SEQUENCE (

```

```

200000, OL_W_ID STARTING FROM 198401 ENDING AT
200000, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE126;
CREATE TABLE ORDER_LINE126
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_126
INDEX IN ts_orderline_126
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 200001 ENDING AT
201600, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE127;
CREATE TABLE ORDER_LINE127
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_127
INDEX IN ts_orderline_127
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 201601 ENDING AT
203200, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE128;

```

```

CREATE TABLE ORDER_LINE128
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_128
INDEX IN ts_orderline_128
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 203201 ENDING AT
204800, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE129;
CREATE TABLE ORDER_LINE129
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_129
INDEX IN ts_orderline_129
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 204801 ENDING AT
206400, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE130;
CREATE TABLE ORDER_LINE130
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,

```

```

  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_130
INDEX IN ts_orderline_130
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 206401 ENDING AT
208000, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE131;
CREATE TABLE ORDER_LINE131
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_131
INDEX IN ts_orderline_131
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 208001 ENDING AT
209600, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE132;
CREATE TABLE ORDER_LINE132
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_132
INDEX IN ts_orderline_132
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 209601 ENDING AT
211200, OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 0 ENDING AT 3694,

```



```

        OL_NUMBER STARTING FROM 1 ENDING AT 15
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE133;
CREATE TABLE ORDER_LINE133
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_133
INDEX IN ts_orderline_133
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 211201 ENDING AT
212800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE134;
CREATE TABLE ORDER_LINE134
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_134
INDEX IN ts_orderline_134
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 212801 ENDING AT
214400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE135;
CREATE TABLE ORDER_LINE135
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,

```

```

    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_135
INDEX IN ts_orderline_135
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 214401 ENDING AT
216000,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE136;
CREATE TABLE ORDER_LINE136
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_136
INDEX IN ts_orderline_136
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 216001 ENDING AT
217600,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE137;
CREATE TABLE ORDER_LINE137
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_137

```

```

INDEX IN ts_orderline_137
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 217601 ENDING AT
219200,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE138;
CREATE TABLE ORDER_LINE138
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_138
INDEX IN ts_orderline_138
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 219201 ENDING AT
220800,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE139;
CREATE TABLE ORDER_LINE139
(
    OL_DELIVERY_D BIGINT NOT NULL,
    OL_AMOUNT INTEGER NOT NULL,
    OL_I_ID INTEGER NOT NULL,
    OL_SUPPLY_W_ID INTEGER NOT NULL,
    OL_QUANTITY SMALLINT NOT NULL,
    OL_DIST_INFO CHAR(24) NOT NULL,
    OL_O_ID INTEGER NOT NULL,
    OL_D_ID SMALLINT NOT NULL,
    OL_W_ID INTEGER NOT NULL,
    OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_139
INDEX IN ts_orderline_139
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 220801 ENDING AT
222400,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 0 ENDING AT 3694,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

```

```

connect to TPCC in share mode;
DROP TABLE ORDER_LINE140;
CREATE TABLE ORDER_LINE140
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_140
INDEX IN ts_orderline_140
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 222401 ENDING AT
224000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE141;
CREATE TABLE ORDER_LINE141
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_141
INDEX IN ts_orderline_141
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 224001 ENDING AT
225600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE142;
CREATE TABLE ORDER_LINE142
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT 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_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_142
INDEX IN ts_orderline_142
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 225601 ENDING AT
227200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE143;
CREATE TABLE ORDER_LINE143
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_143
INDEX IN ts_orderline_143
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 227201 ENDING AT
228800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE144;
CREATE TABLE ORDER_LINE144
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_144
INDEX IN ts_orderline_144
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 228801 ENDING AT
230400,

```

```

  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE145;
CREATE TABLE ORDER_LINE145
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_145
INDEX IN ts_orderline_145
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 230401 ENDING AT
232000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE146;
CREATE TABLE ORDER_LINE146
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_146
INDEX IN ts_orderline_146
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 232001 ENDING AT
233600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE147;
CREATE TABLE ORDER_LINE147
(

```

```

OL_DELIVERY_D BIGINT NOT NULL,
OL_AMOUNT INTEGER NOT NULL,
OL_I_ID INTEGER NOT NULL,
OL_SUPPLY_W_ID INTEGER NOT NULL,
OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_147
INDEX IN ts_orderline_147
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 233601 ENDING AT
235200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE148;
CREATE TABLE ORDER_LINE148
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_148
INDEX IN ts_orderline_148
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 235201 ENDING AT
236800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE149;
CREATE TABLE ORDER_LINE149
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL

```

```

)
IN ts_orderline_149
INDEX IN ts_orderline_149
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 236801 ENDING AT
238400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE150;
CREATE TABLE ORDER_LINE150
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_150
INDEX IN ts_orderline_150
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 238401 ENDING AT
240000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE151;
CREATE TABLE ORDER_LINE151
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_151
INDEX IN ts_orderline_151
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 240001 ENDING AT
241600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE152;
CREATE TABLE ORDER_LINE152
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_152
INDEX IN ts_orderline_152
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 241601 ENDING AT
243200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE153;
CREATE TABLE ORDER_LINE153
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_153
INDEX IN ts_orderline_153
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 243201 ENDING AT
244800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE154;
CREATE TABLE ORDER_LINE154
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,

```

```

OL_QUANTITY SMALLINT NOT NULL,
OL_DIST_INFO CHAR(24) NOT NULL,
OL_O_ID INTEGER NOT NULL,
OL_D_ID SMALLINT NOT NULL,
OL_W_ID INTEGER NOT NULL,
OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_154
INDEX IN ts_orderline_154
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 244801 ENDING AT
246400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE155;
CREATE TABLE ORDER_LINE155
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_155
INDEX IN ts_orderline_155
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 246401 ENDING AT
248000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE156;
CREATE TABLE ORDER_LINE156
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_156
INDEX IN ts_orderline_156
ORGANIZE BY KEY SEQUENCE (

```

```

  OL_W_ID STARTING FROM 248001 ENDING AT
249600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE157;
CREATE TABLE ORDER_LINE157
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_157
INDEX IN ts_orderline_157
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 249601 ENDING AT
251200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE158;
CREATE TABLE ORDER_LINE158
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_158
INDEX IN ts_orderline_158
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 251201 ENDING AT
252800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE159;

```

```

CREATE TABLE ORDER_LINE159
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_159
INDEX IN ts_orderline_159
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 252801 ENDING AT
254400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE160;
CREATE TABLE ORDER_LINE160
(
  OL_DELIVERY_D BIGINT NOT NULL,
  OL_AMOUNT INTEGER NOT NULL,
  OL_I_ID INTEGER NOT NULL,
  OL_SUPPLY_W_ID INTEGER NOT NULL,
  OL_QUANTITY SMALLINT NOT NULL,
  OL_DIST_INFO CHAR(24) NOT NULL,
  OL_O_ID INTEGER NOT NULL,
  OL_D_ID SMALLINT NOT NULL,
  OL_W_ID INTEGER NOT NULL,
  OL_NUMBER SMALLINT NOT NULL
)
IN ts_orderline_160
INDEX IN ts_orderline_160
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 254401 ENDING AT
256000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 0 ENDING AT 3694,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;

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 1600
)
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 1601 ENDING AT
3200
)
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 3201 ENDING AT
4800
)
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 4801 ENDING AT
6400
)
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 6401 ENDING AT
8000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK6;
CREATE TABLE STOCK6
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24) NOT NULL,
S_DIST_02   CHAR(24) NOT NULL,
S_DIST_03   CHAR(24) NOT NULL,
S_DIST_04   CHAR(24) NOT NULL,
S_DIST_05   CHAR(24) NOT NULL,
S_DIST_06   CHAR(24) NOT NULL,
S_DIST_07   CHAR(24) NOT NULL,
S_DIST_08   CHAR(24) NOT NULL,
S_DIST_09   CHAR(24) NOT NULL,
S_DIST_10   CHAR(24) NOT NULL,
S_I_ID      INTEGER NOT NULL,
S_W_ID      INTEGER NOT NULL
)
IN ts_stock_006
INDEX IN ts_stock_006
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 8001 ENDING AT
9600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK7;
CREATE TABLE STOCK7
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_007
INDEX IN ts_stock_007
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 9601 ENDING AT
11200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK8;
CREATE TABLE STOCK8
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_008
INDEX IN ts_stock_008
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 11201 ENDING AT
12800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK9;
CREATE TABLE STOCK9
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_009
INDEX IN ts_stock_009
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 12801 ENDING AT
14400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK10;
CREATE TABLE STOCK10
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_010
INDEX IN ts_stock_010
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 14401 ENDING AT
16000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK11;
CREATE TABLE STOCK11
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_011
INDEX IN ts_stock_011
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 16001 ENDING AT
17600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK12;
CREATE TABLE STOCK12
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_012
INDEX IN ts_stock_012
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 17601 ENDING AT
19200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK13;
CREATE TABLE STOCK13
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_013
INDEX IN ts_stock_013
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 19201 ENDING AT
20800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK14;
CREATE TABLE STOCK14
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_014
INDEX IN ts_stock_014
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 20801 ENDING AT
22400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK15;
CREATE TABLE STOCK15
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_015
INDEX IN ts_stock_015
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 22401 ENDING AT
24000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK16;
CREATE TABLE STOCK16
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_016
INDEX IN ts_stock_016
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 24001 ENDING AT
25600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK17;
CREATE TABLE STOCK17
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_017
INDEX IN ts_stock_017
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 25601 ENDING AT
27200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK18;
CREATE TABLE STOCK18
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_018
INDEX IN ts_stock_018
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 27201 ENDING AT
28800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK19;
CREATE TABLE STOCK19
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_019
INDEX IN ts_stock_019
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 28801 ENDING AT
30400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK20;
CREATE TABLE STOCK20
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_020
INDEX IN ts_stock_020
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 30401 ENDING AT
32000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK21;
CREATE TABLE STOCK21
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_021
INDEX IN ts_stock_021
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 32001 ENDING AT
33600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK22;
CREATE TABLE STOCK22
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_022
INDEX IN ts_stock_022
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 33601 ENDING AT
35200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK23;
CREATE TABLE STOCK23
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_023
INDEX IN ts_stock_023
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 35201 ENDING AT
36800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK24;
CREATE TABLE STOCK24
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_024
INDEX IN ts_stock_024
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 36801 ENDING AT
38400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK25;
CREATE TABLE STOCK25
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```



```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_025
INDEX IN ts_stock_025
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 38401 ENDING AT
40000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK26;
CREATE TABLE STOCK26
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_026
INDEX IN ts_stock_026
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 40001 ENDING AT
41600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK27;
CREATE TABLE STOCK27
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_027
INDEX IN ts_stock_027
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 41601 ENDING AT
43200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK28;
CREATE TABLE STOCK28
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_028
INDEX IN ts_stock_028
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 43201 ENDING AT
44800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK29;
CREATE TABLE STOCK29
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_029
INDEX IN ts_stock_029
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 44801 ENDING AT
46400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK30;
CREATE TABLE STOCK30
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_030
INDEX IN ts_stock_030
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 46401 ENDING AT
48000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK31;
CREATE TABLE STOCK31
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_031
INDEX IN ts_stock_031
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 48001 ENDING AT
49600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK32;
CREATE TABLE STOCK32
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_032
INDEX IN ts_stock_032
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 49601 ENDING AT
51200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK33;
CREATE TABLE STOCK33
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_033
INDEX IN ts_stock_033
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 51201 ENDING AT
52800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK34;
CREATE TABLE STOCK34
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_034
INDEX IN ts_stock_034
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 52801 ENDING AT
54400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK35;
CREATE TABLE STOCK35
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_035
INDEX IN ts_stock_035
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 54401 ENDING AT
56000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK36;
CREATE TABLE STOCK36
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_036
INDEX IN ts_stock_036
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 56001 ENDING AT
57600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK37;
CREATE TABLE STOCK37
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_037
INDEX IN ts_stock_037
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 57601 ENDING AT
59200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK38;
CREATE TABLE STOCK38
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_038
INDEX IN ts_stock_038
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 59201 ENDING AT
60800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK39;
CREATE TABLE STOCK39
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_039
INDEX IN ts_stock_039
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 60801 ENDING AT
62400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK40;
CREATE TABLE STOCK40
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_040
INDEX IN ts_stock_040
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 62401 ENDING AT
64000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK41;
CREATE TABLE STOCK41
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_041
INDEX IN ts_stock_041
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 64001 ENDING AT
65600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK42;
CREATE TABLE STOCK42
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_042
INDEX IN ts_stock_042
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 65601 ENDING AT
67200
)
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 67201 ENDING AT
68800
)
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 68801 ENDING AT
70400
)
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 70401 ENDING AT
72000
)
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 72001 ENDING AT
73600
)
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 73601 ENDING AT
75200
)
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 75201 ENDING AT
76800
)
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 76801 ENDING AT
78400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK50;
CREATE TABLE STOCK50
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_050
INDEX IN ts_stock_050
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 78401 ENDING AT
80000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK51;
CREATE TABLE STOCK51
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_051
INDEX IN ts_stock_051
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 80001 ENDING AT
81600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK52;
CREATE TABLE STOCK52
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_052
INDEX IN ts_stock_052
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 81601 ENDING AT
83200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK53;
CREATE TABLE STOCK53
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_053
INDEX IN ts_stock_053
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 83201 ENDING AT
84800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK54;
CREATE TABLE STOCK54
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_054
INDEX IN ts_stock_054
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 84801 ENDING AT
86400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK55;
CREATE TABLE STOCK55
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_055
INDEX IN ts_stock_055
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 86401 ENDING AT
88000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK56;
CREATE TABLE STOCK56
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24)  NOT NULL,
S_DIST_02    CHAR(24)  NOT NULL,
S_DIST_03    CHAR(24)  NOT NULL,
S_DIST_04    CHAR(24)  NOT NULL,
S_DIST_05    CHAR(24)  NOT NULL,
S_DIST_06    CHAR(24)  NOT NULL,
S_DIST_07    CHAR(24)  NOT NULL,
S_DIST_08    CHAR(24)  NOT NULL,
S_DIST_09    CHAR(24)  NOT NULL,
S_DIST_10    CHAR(24)  NOT NULL,
S_I_ID       INTEGER   NOT NULL,
S_W_ID       INTEGER   NOT NULL
)
IN ts_stock_056
INDEX IN ts_stock_056
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 88001 ENDING AT
89600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK57;
CREATE TABLE STOCK57
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_057
INDEX IN ts_stock_057
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 89601 ENDING AT
91200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK58;
CREATE TABLE STOCK58
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24)  NOT NULL,
S_DIST_02    CHAR(24)  NOT NULL,
S_DIST_03    CHAR(24)  NOT NULL,
S_DIST_04    CHAR(24)  NOT NULL,
S_DIST_05    CHAR(24)  NOT NULL,
S_DIST_06    CHAR(24)  NOT NULL,
S_DIST_07    CHAR(24)  NOT NULL,
S_DIST_08    CHAR(24)  NOT NULL,
S_DIST_09    CHAR(24)  NOT NULL,
S_DIST_10    CHAR(24)  NOT NULL,
S_I_ID       INTEGER   NOT NULL,
S_W_ID       INTEGER   NOT NULL
)
IN ts_stock_058
INDEX IN ts_stock_058
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 91201 ENDING AT
92800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK59;
CREATE TABLE STOCK59
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_059
INDEX IN ts_stock_059
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 92801 ENDING AT
94400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK60;
CREATE TABLE STOCK60
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24)  NOT NULL,
S_DIST_02    CHAR(24)  NOT NULL,
S_DIST_03    CHAR(24)  NOT NULL,
S_DIST_04    CHAR(24)  NOT NULL,
S_DIST_05    CHAR(24)  NOT NULL,
S_DIST_06    CHAR(24)  NOT NULL,
S_DIST_07    CHAR(24)  NOT NULL,
S_DIST_08    CHAR(24)  NOT NULL,
S_DIST_09    CHAR(24)  NOT NULL,
S_DIST_10    CHAR(24)  NOT NULL,
S_I_ID       INTEGER   NOT NULL,
S_W_ID       INTEGER   NOT NULL
)
IN ts_stock_060
INDEX IN ts_stock_060
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 94401 ENDING AT
96000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK61;
CREATE TABLE STOCK61
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY   INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_061
INDEX IN ts_stock_061
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 96001 ENDING AT
97600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK62;
CREATE TABLE STOCK62
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_062
INDEX IN ts_stock_062
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 97601 ENDING AT
99200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK63;
CREATE TABLE STOCK63
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_063
INDEX IN ts_stock_063
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 99201 ENDING AT
100800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK64;
CREATE TABLE STOCK64
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_064
INDEX IN ts_stock_064
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 100801 ENDING AT
102400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK65;
CREATE TABLE STOCK65
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_065
INDEX IN ts_stock_065
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 102401 ENDING AT
104000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK66;
CREATE TABLE STOCK66
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_066
INDEX IN ts_stock_066
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 104001 ENDING AT
105600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK67;
CREATE TABLE STOCK67
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_067
INDEX IN ts_stock_067
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 105601 ENDING AT
107200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK68;
CREATE TABLE STOCK68
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24)  NOT NULL,
S_DIST_02    CHAR(24)  NOT NULL,
S_DIST_03    CHAR(24)  NOT NULL,
S_DIST_04    CHAR(24)  NOT NULL,
S_DIST_05    CHAR(24)  NOT NULL,
S_DIST_06    CHAR(24)  NOT NULL,
S_DIST_07    CHAR(24)  NOT NULL,
S_DIST_08    CHAR(24)  NOT NULL,
S_DIST_09    CHAR(24)  NOT NULL,
S_DIST_10    CHAR(24)  NOT NULL,
S_I_ID       INTEGER   NOT NULL,
S_W_ID       INTEGER   NOT NULL
)
IN ts_stock_068
INDEX IN ts_stock_068
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 107201 ENDING AT
108800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK69;
CREATE TABLE STOCK69
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_069
INDEX IN ts_stock_069
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 108801 ENDING AT
110400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK70;
CREATE TABLE STOCK70
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24)  NOT NULL,
S_DIST_02    CHAR(24)  NOT NULL,
S_DIST_03    CHAR(24)  NOT NULL,
S_DIST_04    CHAR(24)  NOT NULL,
S_DIST_05    CHAR(24)  NOT NULL,
S_DIST_06    CHAR(24)  NOT NULL,
S_DIST_07    CHAR(24)  NOT NULL,
S_DIST_08    CHAR(24)  NOT NULL,
S_DIST_09    CHAR(24)  NOT NULL,
S_DIST_10    CHAR(24)  NOT NULL,
S_I_ID       INTEGER   NOT NULL,
S_W_ID       INTEGER   NOT NULL
)
IN ts_stock_070
INDEX IN ts_stock_070
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 110401 ENDING AT
112000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK71;
CREATE TABLE STOCK71
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_071
INDEX IN ts_stock_071
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 112001 ENDING AT
113600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK72;
CREATE TABLE STOCK72
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01    CHAR(24)  NOT NULL,
S_DIST_02    CHAR(24)  NOT NULL,
S_DIST_03    CHAR(24)  NOT NULL,
S_DIST_04    CHAR(24)  NOT NULL,
S_DIST_05    CHAR(24)  NOT NULL,
S_DIST_06    CHAR(24)  NOT NULL,
S_DIST_07    CHAR(24)  NOT NULL,
S_DIST_08    CHAR(24)  NOT NULL,
S_DIST_09    CHAR(24)  NOT NULL,
S_DIST_10    CHAR(24)  NOT NULL,
S_I_ID       INTEGER   NOT NULL,
S_W_ID       INTEGER   NOT NULL
)
IN ts_stock_072
INDEX IN ts_stock_072
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 113601 ENDING AT
115200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK73;
CREATE TABLE STOCK73
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```



```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_073
INDEX IN ts_stock_073
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 115201 ENDING AT
116800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK74;
CREATE TABLE STOCK74
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_074
INDEX IN ts_stock_074
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 116801 ENDING AT
118400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK75;
CREATE TABLE STOCK75
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_075
INDEX IN ts_stock_075
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 118401 ENDING AT
120000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK76;
CREATE TABLE STOCK76
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_076
INDEX IN ts_stock_076
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 120001 ENDING AT
121600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK77;
CREATE TABLE STOCK77
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_077
INDEX IN ts_stock_077
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 121601 ENDING AT
123200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK78;
CREATE TABLE STOCK78
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,
S_DATA       VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_078
INDEX IN ts_stock_078
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 123201 ENDING AT
124800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK79;
CREATE TABLE STOCK79
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT  INTEGER NOT NULL,
S_YTD        INTEGER NOT NULL,

```

```

S_DATA    VARCHAR(50) NOT NULL,
S_DIST_01 CHAR(24)  NOT NULL,
S_DIST_02 CHAR(24)  NOT NULL,
S_DIST_03 CHAR(24)  NOT NULL,
S_DIST_04 CHAR(24)  NOT NULL,
S_DIST_05 CHAR(24)  NOT NULL,
S_DIST_06 CHAR(24)  NOT NULL,
S_DIST_07 CHAR(24)  NOT NULL,
S_DIST_08 CHAR(24)  NOT NULL,
S_DIST_09 CHAR(24)  NOT NULL,
S_DIST_10 CHAR(24)  NOT NULL,
S_I_ID    INTEGER   NOT NULL,
S_W_ID    INTEGER   NOT NULL
)
IN ts_stock_079
INDEX IN ts_stock_079
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 124801 ENDING AT
126400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK80;
CREATE TABLE STOCK80
(
S_REMOTE_CNT INTEGER NOT NULL,
S_QUANTITY  INTEGER NOT NULL,
S_ORDER_CNT INTEGER NOT NULL,
S_YTD       INTEGER NOT NULL,
S_DATA      VARCHAR(50) NOT NULL,
S_DIST_01   CHAR(24)  NOT NULL,
S_DIST_02   CHAR(24)  NOT NULL,
S_DIST_03   CHAR(24)  NOT NULL,
S_DIST_04   CHAR(24)  NOT NULL,
S_DIST_05   CHAR(24)  NOT NULL,
S_DIST_06   CHAR(24)  NOT NULL,
S_DIST_07   CHAR(24)  NOT NULL,
S_DIST_08   CHAR(24)  NOT NULL,
S_DIST_09   CHAR(24)  NOT NULL,
S_DIST_10   CHAR(24)  NOT NULL,
S_I_ID      INTEGER   NOT NULL,
S_W_ID      INTEGER   NOT NULL
)
IN ts_stock_080
INDEX IN ts_stock_080
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 126401 ENDING AT
128000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK81;
CREATE TABLE STOCK81
(
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_081
INDEX IN ts_stock_081
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 128001 ENDING AT
129600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK82;
CREATE TABLE STOCK82
(
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_082
INDEX IN ts_stock_082
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 129601 ENDING AT
131200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK83;
CREATE TABLE STOCK83
(
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_083
INDEX IN ts_stock_083
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 131201 ENDING AT
132800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK84;
CREATE TABLE STOCK84
(
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_084
INDEX IN ts_stock_084
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 132801 ENDING AT
134400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK85;
CREATE TABLE STOCK85
(
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_085
INDEX IN ts_stock_085
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 134401 ENDING AT
136000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK86;
CREATE TABLE STOCK86
(
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_086
INDEX IN ts_stock_086
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 136001 ENDING AT
137600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK87;
CREATE TABLE STOCK87
(
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_087
INDEX IN ts_stock_087
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 137601 ENDING AT
139200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK88;
CREATE TABLE STOCK88
(
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_088
INDEX IN ts_stock_088
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 139201 ENDING AT
140800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK89;
CREATE TABLE STOCK89
(
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_089
INDEX IN ts_stock_089
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 140801 ENDING AT
142400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK90;
CREATE TABLE STOCK90
(
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_090
INDEX IN ts_stock_090
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 142401 ENDING AT
144000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK91;
CREATE TABLE STOCK91
(
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_091
INDEX IN ts_stock_091
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 144001 ENDING AT
145600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK92;
CREATE TABLE STOCK92
(
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_092
INDEX IN ts_stock_092
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 145601 ENDING AT
147200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK93;
CREATE TABLE STOCK93
(
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_093
INDEX IN ts_stock_093
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 147201 ENDING AT
148800
)
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 148801 ENDING AT
150400
)
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 150401 ENDING AT
152000
)
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 152001 ENDING AT
153600
)
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 153601 ENDING AT
155200
)
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 155201 ENDING AT
156800
)
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 156801 ENDING AT
158400
)
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 158401 ENDING AT
160000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK101;
CREATE TABLE STOCK101
(
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_101
INDEX IN ts_stock_101
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 160001 ENDING AT
161600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK102;
CREATE TABLE STOCK102
(
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_102
INDEX IN ts_stock_102
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 161601 ENDING AT
163200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK103;
CREATE TABLE STOCK103
(
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_103
INDEX IN ts_stock_103
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 163201 ENDING AT
164800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK104;
CREATE TABLE STOCK104
(
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_104
INDEX IN ts_stock_104
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 164801 ENDING AT
166400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK105;
CREATE TABLE STOCK105
(
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_105
INDEX IN ts_stock_105
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 166401 ENDING AT
168000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK106;
CREATE TABLE STOCK106
(
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_106
INDEX IN ts_stock_106
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 168001 ENDING AT
169600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK107;
CREATE TABLE STOCK107
(
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_107
INDEX IN ts_stock_107
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 169601 ENDING AT
171200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK108;
CREATE TABLE STOCK108
(
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_108
INDEX IN ts_stock_108
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 171201 ENDING AT
172800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK109;
CREATE TABLE STOCK109
(
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_109
INDEX IN ts_stock_109
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 172801 ENDING AT
174400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK110;
CREATE TABLE STOCK110
(
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_110
INDEX IN ts_stock_110
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 174401 ENDING AT
176000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK111;
CREATE TABLE STOCK111
(
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_111
INDEX IN ts_stock_111
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 176001 ENDING AT
177600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK112;
CREATE TABLE STOCK112
(
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_112
INDEX IN ts_stock_112
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 177601 ENDING AT
179200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK113;
CREATE TABLE STOCK113
(
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_113
INDEX IN ts_stock_113
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 179201 ENDING AT
180800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK114;
CREATE TABLE STOCK114
(
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_114
INDEX IN ts_stock_114
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 180801 ENDING AT
182400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK115;
CREATE TABLE STOCK115
(
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_115
INDEX IN ts_stock_115
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 182401 ENDING AT
184000
)
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 184001 ENDING AT
185600
)
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 185601 ENDING AT
187200
)
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 187201 ENDING AT
188800
)
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 188801 ENDING AT
190400
)
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 190401 ENDING AT
192000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK121;
CREATE TABLE STOCK121
(
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_121
INDEX IN ts_stock_121
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 192001 ENDING AT
193600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK122;
CREATE TABLE STOCK122
(
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_122
INDEX IN ts_stock_122
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 193601 ENDING AT
195200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK123;
CREATE TABLE STOCK123
(
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_123
INDEX IN ts_stock_123
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 195201 ENDING AT
196800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK124;
CREATE TABLE STOCK124
(
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_124
INDEX IN ts_stock_124
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 196801 ENDING AT
198400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK125;
CREATE TABLE STOCK125
(
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_125
INDEX IN ts_stock_125
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 198401 ENDING AT
200000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK126;
CREATE TABLE STOCK126
(
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_126
INDEX IN ts_stock_126
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 200001 ENDING AT
201600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK127;
CREATE TABLE STOCK127
(
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_127
INDEX IN ts_stock_127
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 201601 ENDING AT
203200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK128;
CREATE TABLE STOCK128
(
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_128
INDEX IN ts_stock_128
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 203201 ENDING AT
204800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK129;
CREATE TABLE STOCK129
(
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_129
INDEX IN ts_stock_129
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 204801 ENDING AT
206400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK130;
CREATE TABLE STOCK130
(
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_130
INDEX IN ts_stock_130
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 206401 ENDING AT
208000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK131;
CREATE TABLE STOCK131
(
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_131
INDEX IN ts_stock_131
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 208001 ENDING AT
209600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK132;
CREATE TABLE STOCK132
(
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_132
INDEX IN ts_stock_132
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 209601 ENDING AT
211200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK133;
CREATE TABLE STOCK133
(
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_133
INDEX IN ts_stock_133
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 211201 ENDING AT
212800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK134;
CREATE TABLE STOCK134
(
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_134
INDEX IN ts_stock_134
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 212801 ENDING AT
214400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK135;
CREATE TABLE STOCK135
(
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_135
INDEX IN ts_stock_135
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 214401 ENDING AT
216000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK136;
CREATE TABLE STOCK136
(
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_136
INDEX IN ts_stock_136
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 216001 ENDING AT
217600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK137;
CREATE TABLE STOCK137
(
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_137
INDEX IN ts_stock_137
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 217601 ENDING AT
219200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK138;
CREATE TABLE STOCK138
(
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_138
INDEX IN ts_stock_138
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 219201 ENDING AT
220800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK139;
CREATE TABLE STOCK139
(
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_139
INDEX IN ts_stock_139
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 220801 ENDING AT
222400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK140;
CREATE TABLE STOCK140
(
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_140
INDEX IN ts_stock_140
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 222401 ENDING AT
224000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK141;
CREATE TABLE STOCK141
(
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_141
INDEX IN ts_stock_141
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 224001 ENDING AT
225600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK142;
CREATE TABLE STOCK142
(
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_142
INDEX IN ts_stock_142
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 225601 ENDING AT
227200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK143;
CREATE TABLE STOCK143
(
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_143
INDEX IN ts_stock_143
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 227201 ENDING AT
228800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK144;
CREATE TABLE STOCK144
(
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_144
INDEX IN ts_stock_144
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 228801 ENDING AT
230400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK145;
CREATE TABLE STOCK145
(
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_145
INDEX IN ts_stock_145
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 230401 ENDING AT
232000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK146;
CREATE TABLE STOCK146
(
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_146
INDEX IN ts_stock_146
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 232001 ENDING AT
233600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK147;
CREATE TABLE STOCK147
(
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_147
INDEX IN ts_stock_147
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 233601 ENDING AT
235200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK148;
CREATE TABLE STOCK148
(
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_148
INDEX IN ts_stock_148
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 235201 ENDING AT
236800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK149;
CREATE TABLE STOCK149
(
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_149
INDEX IN ts_stock_149
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 236801 ENDING AT
238400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK150;
CREATE TABLE STOCK150
(
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_150
INDEX IN ts_stock_150
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 238401 ENDING AT
240000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK151;
CREATE TABLE STOCK151
(
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_151
INDEX IN ts_stock_151
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 240001 ENDING AT
241600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK152;
CREATE TABLE STOCK152
(
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_152
INDEX IN ts_stock_152
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 241601 ENDING AT
243200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK153;
CREATE TABLE STOCK153
(
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_153
INDEX IN ts_stock_153
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 243201 ENDING AT
244800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK154;
CREATE TABLE STOCK154
(
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_154
INDEX IN ts_stock_154
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 244801 ENDING AT
246400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK155;
CREATE TABLE STOCK155
(
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_155
INDEX IN ts_stock_155
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 246401 ENDING AT
248000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK156;
CREATE TABLE STOCK156
(
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_156
INDEX IN ts_stock_156
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 248001 ENDING AT
249600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK157;
CREATE TABLE STOCK157
(
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_157
INDEX IN ts_stock_157
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 249601 ENDING AT
251200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK158;
CREATE TABLE STOCK158
(
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_158
INDEX IN ts_stock_158
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 251201 ENDING AT
252800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK159;
CREATE TABLE STOCK159
(
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_159
INDEX IN ts_stock_159
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 252801 ENDING AT
254400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK160;
CREATE TABLE STOCK160
(
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_160
INDEX IN ts_stock_160
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 254401 ENDING AT
256000
)
ALLOW OVERFLOW;
connect reset;

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 INTEGER NOT NULL,
W_YTD BIGINT 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 32000
)
ALLOW OVERFLOW
PCTFREE 50;
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 INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_002
INDEX IN ts_ware_002
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 32001 ENDING AT 64000
)
ALLOW OVERFLOW
PCTFREE 50;
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 INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_003
INDEX IN ts_ware_003
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 64001 ENDING AT 96000
)
ALLOW OVERFLOW
PCTFREE 50;

```

```

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 INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_004
INDEX IN ts_ware_004
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 96001 ENDING AT
128000
)
ALLOW OVERFLOW
PCTFREE 50;

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 INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_005
INDEX IN ts_ware_005
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 128001 ENDING AT
160000
)
ALLOW OVERFLOW
PCTFREE 50;

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 INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_006

```

```

INDEX IN ts_ware_006
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 160001 ENDING AT
192000
)
ALLOW OVERFLOW
PCTFREE 50;

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 INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_007
INDEX IN ts_ware_007
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 192001 ENDING AT
224000
)
ALLOW OVERFLOW
PCTFREE 50;

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 INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_008
INDEX IN ts_ware_008
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 224001 ENDING AT
256000
)
ALLOW OVERFLOW
PCTFREE 50;

connect reset;

CRVW CUSTOMER.ddl

connect to TPCC in share mode;
DROP VIEW CUSTOMER;
CREATE VIEW CUSTOMER

```

```

(C_ID,
C_STATE,
C_ZIP,
C_PHONE,
C_SINCE,
C_CREDIT_LIM,
C_MIDDLE,
C_CREDIT,
C_DISCOUNT,
C_DATA,
C_LAST,
C_FIRST,
C_STREET_1,
C_STREET_2,
C_CITY,
C_D_ID,
C_W_ID,
C_DELIVERY_CNT,
C_BALANCE,
C_YTD_PAYMENT,
C_PAYMENT_CNT
) AS SELECT * FROM CUSTOMER1 UNION ALL
SELECT * FROM CUSTOMER2 UNION ALL
SELECT * FROM CUSTOMER3 UNION ALL
SELECT * FROM CUSTOMER4 UNION ALL
SELECT * FROM CUSTOMER5 UNION ALL
SELECT * FROM CUSTOMER6 UNION ALL
SELECT * FROM CUSTOMER7 UNION ALL
SELECT * FROM CUSTOMER8 UNION ALL
SELECT * FROM CUSTOMER9 UNION ALL
SELECT * FROM CUSTOMER10 UNION ALL
SELECT * FROM CUSTOMER11 UNION ALL
SELECT * FROM CUSTOMER12 UNION ALL
SELECT * FROM CUSTOMER13 UNION ALL
SELECT * FROM CUSTOMER14 UNION ALL
SELECT * FROM CUSTOMER15 UNION ALL
SELECT * FROM CUSTOMER16 UNION ALL
SELECT * FROM CUSTOMER17 UNION ALL
SELECT * FROM CUSTOMER18 UNION ALL
SELECT * FROM CUSTOMER19 UNION ALL
SELECT * FROM CUSTOMER20 UNION ALL
SELECT * FROM CUSTOMER21 UNION ALL
SELECT * FROM CUSTOMER22 UNION ALL
SELECT * FROM CUSTOMER23 UNION ALL
SELECT * FROM CUSTOMER24 UNION ALL
SELECT * FROM CUSTOMER25 UNION ALL
SELECT * FROM CUSTOMER26 UNION ALL
SELECT * FROM CUSTOMER27 UNION ALL
SELECT * FROM CUSTOMER28 UNION ALL
SELECT * FROM CUSTOMER29 UNION ALL
SELECT * FROM CUSTOMER30 UNION ALL
SELECT * FROM CUSTOMER31 UNION ALL
SELECT * FROM CUSTOMER32 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38 UNION ALL
SELECT * FROM CUSTOMER39 UNION ALL
SELECT * FROM CUSTOMER40 UNION ALL
SELECT * FROM CUSTOMER41 UNION ALL

```



```

SELECT * FROM CUSTOMER42 UNION ALL
SELECT * FROM CUSTOMER43 UNION ALL
SELECT * FROM CUSTOMER44 UNION ALL
SELECT * FROM CUSTOMER45 UNION ALL
SELECT * FROM CUSTOMER46 UNION ALL
SELECT * FROM CUSTOMER47 UNION ALL
SELECT * FROM CUSTOMER48 UNION ALL
SELECT * FROM CUSTOMER49 UNION ALL
SELECT * FROM CUSTOMER50 UNION ALL
SELECT * FROM CUSTOMER51 UNION ALL
SELECT * FROM CUSTOMER52 UNION ALL
SELECT * FROM CUSTOMER53 UNION ALL
SELECT * FROM CUSTOMER54 UNION ALL
SELECT * FROM CUSTOMER55 UNION ALL
SELECT * FROM CUSTOMER56 UNION ALL
SELECT * FROM CUSTOMER57 UNION ALL
SELECT * FROM CUSTOMER58 UNION ALL
SELECT * FROM CUSTOMER59 UNION ALL
SELECT * FROM CUSTOMER60 UNION ALL
SELECT * FROM CUSTOMER61 UNION ALL
SELECT * FROM CUSTOMER62 UNION ALL
SELECT * FROM CUSTOMER63 UNION ALL
SELECT * FROM CUSTOMER64 UNION ALL
SELECT * FROM CUSTOMER65 UNION ALL
SELECT * FROM CUSTOMER66 UNION ALL
SELECT * FROM CUSTOMER67 UNION ALL
SELECT * FROM CUSTOMER68 UNION ALL
SELECT * FROM CUSTOMER69 UNION ALL
SELECT * FROM CUSTOMER70 UNION ALL
SELECT * FROM CUSTOMER71 UNION ALL
SELECT * FROM CUSTOMER72 UNION ALL
SELECT * FROM CUSTOMER73 UNION ALL
SELECT * FROM CUSTOMER74 UNION ALL
SELECT * FROM CUSTOMER75 UNION ALL
SELECT * FROM CUSTOMER76 UNION ALL
SELECT * FROM CUSTOMER77 UNION ALL
SELECT * FROM CUSTOMER78 UNION ALL
SELECT * FROM CUSTOMER79 UNION ALL
SELECT * FROM CUSTOMER80 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 UNION ALL
SELECT * FROM CUSTOMER121 UNION ALL
SELECT * FROM CUSTOMER122 UNION ALL
SELECT * FROM CUSTOMER123 UNION ALL
SELECT * FROM CUSTOMER124 UNION ALL
SELECT * FROM CUSTOMER125 UNION ALL
SELECT * FROM CUSTOMER126 UNION ALL
SELECT * FROM CUSTOMER127 UNION ALL
SELECT * FROM CUSTOMER128 UNION ALL
SELECT * FROM CUSTOMER129 UNION ALL
SELECT * FROM CUSTOMER130 UNION ALL
SELECT * FROM CUSTOMER131 UNION ALL
SELECT * FROM CUSTOMER132 UNION ALL
SELECT * FROM CUSTOMER133 UNION ALL
SELECT * FROM CUSTOMER134 UNION ALL
SELECT * FROM CUSTOMER135 UNION ALL
SELECT * FROM CUSTOMER136 UNION ALL
SELECT * FROM CUSTOMER137 UNION ALL
SELECT * FROM CUSTOMER138 UNION ALL
SELECT * FROM CUSTOMER139 UNION ALL
SELECT * FROM CUSTOMER140 UNION ALL
SELECT * FROM CUSTOMER141 UNION ALL
SELECT * FROM CUSTOMER142 UNION ALL
SELECT * FROM CUSTOMER143 UNION ALL
SELECT * FROM CUSTOMER144 UNION ALL
SELECT * FROM CUSTOMER145 UNION ALL
SELECT * FROM CUSTOMER146 UNION ALL
SELECT * FROM CUSTOMER147 UNION ALL
SELECT * FROM CUSTOMER148 UNION ALL
SELECT * FROM CUSTOMER149 UNION ALL
SELECT * FROM CUSTOMER150 UNION ALL
SELECT * FROM CUSTOMER151 UNION ALL
SELECT * FROM CUSTOMER152 UNION ALL
SELECT * FROM CUSTOMER153 UNION ALL
SELECT * FROM CUSTOMER154 UNION ALL
SELECT * FROM CUSTOMER155 UNION ALL
SELECT * FROM CUSTOMER156 UNION ALL
SELECT * FROM CUSTOMER157 UNION ALL
SELECT * FROM CUSTOMER158 UNION ALL
SELECT * FROM CUSTOMER159 UNION ALL
SELECT * FROM CUSTOMER160
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW_DISTRICT.ddl

connect to TPCC in share mode;

```

DROP VIEW DISTRICT;
CREATE VIEW DISTRICT
(D_NEXT_O_ID,
D_TAX,
D_YTD,
D_NAME,
D_STREET_1,
D_STREET_2,
D_CITY,
D_STATE,
D_ZIP,
D_ID,
D_W_ID
) AS SELECT * FROM DISTRICT1 UNION ALL
SELECT * FROM DISTRICT2 UNION ALL
SELECT * FROM DISTRICT3 UNION ALL
SELECT * FROM DISTRICT4 UNION ALL
SELECT * FROM DISTRICT5 UNION ALL
SELECT * FROM DISTRICT6 UNION ALL
SELECT * FROM DISTRICT7 UNION ALL
SELECT * FROM DISTRICT8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW_HISTORY.ddl

connect to TPCC in share mode;

```

DROP VIEW HISTORY;
CREATE VIEW HISTORY
(H_C_ID,
H_C_D_ID,
H_C_W_ID,
H_D_ID,
H_W_ID,
H_DATE,
H_AMOUNT,
H_DATA
) AS SELECT * FROM HISTORY1 UNION ALL
SELECT * FROM HISTORY2 UNION ALL
SELECT * FROM HISTORY3 UNION ALL
SELECT * FROM HISTORY4 UNION ALL
SELECT * FROM HISTORY5 UNION ALL
SELECT * FROM HISTORY6 UNION ALL
SELECT * FROM HISTORY7 UNION ALL
SELECT * FROM HISTORY8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW_NEW_ORDER.ddl

connect to TPCC in share mode;
DROP VIEW NEW_ORDER;

```

CREATE VIEW NEW_ORDER
(NO_O_ID,
NO_D_ID,
NO_W_ID
) AS SELECT * FROM NEW_ORDERA1 UNION
ALL
SELECT * FROM NEW_ORDERA2 UNION ALL
SELECT * FROM NEW_ORDERA3 UNION ALL
SELECT * FROM NEW_ORDERA4 UNION ALL
SELECT * FROM NEW_ORDERA5 UNION ALL
SELECT * FROM NEW_ORDERA6 UNION ALL
SELECT * FROM NEW_ORDERA7 UNION ALL
SELECT * FROM NEW_ORDERA8 UNION ALL
SELECT * FROM NEW_ORDERB1 UNION ALL
SELECT * FROM NEW_ORDERB2 UNION ALL
SELECT * FROM NEW_ORDERB3 UNION ALL
SELECT * FROM NEW_ORDERB4 UNION ALL
SELECT * FROM NEW_ORDERB5 UNION ALL
SELECT * FROM NEW_ORDERB6 UNION ALL
SELECT * FROM NEW_ORDERB7 UNION ALL
SELECT * FROM NEW_ORDERB8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW ORDERS.ddl

```

connect to TPCC in share mode;
DROP VIEW ORDERS;
CREATE VIEW ORDERS
(O_C_ID,
O_ENTRY_D,
O_CARRIER_ID,
O_OL_CNT,
O_ALL_LOCAL,
O_ID,
O_W_ID,
O_D_ID
) AS SELECT * FROM ORDERS1 UNION ALL
SELECT * FROM ORDERS2 UNION ALL
SELECT * FROM ORDERS3 UNION ALL
SELECT * FROM ORDERS4 UNION ALL
SELECT * FROM ORDERS5 UNION ALL
SELECT * FROM ORDERS6 UNION ALL
SELECT * FROM ORDERS7 UNION ALL
SELECT * FROM ORDERS8 UNION ALL
SELECT * FROM ORDERS9 UNION ALL
SELECT * FROM ORDERS10 UNION ALL
SELECT * FROM ORDERS11 UNION ALL
SELECT * FROM ORDERS12 UNION ALL
SELECT * FROM ORDERS13 UNION ALL
SELECT * FROM ORDERS14 UNION ALL
SELECT * FROM ORDERS15 UNION ALL
SELECT * FROM ORDERS16 UNION ALL
SELECT * FROM ORDERS17 UNION ALL
SELECT * FROM ORDERS18 UNION ALL
SELECT * FROM ORDERS19 UNION ALL
SELECT * FROM ORDERS20 UNION ALL
SELECT * FROM ORDERS21 UNION ALL
SELECT * FROM ORDERS22 UNION ALL

```

```

SELECT * FROM ORDERS23 UNION ALL
SELECT * FROM ORDERS24 UNION ALL
SELECT * FROM ORDERS25 UNION ALL
SELECT * FROM ORDERS26 UNION ALL
SELECT * FROM ORDERS27 UNION ALL
SELECT * FROM ORDERS28 UNION ALL
SELECT * FROM ORDERS29 UNION ALL
SELECT * FROM ORDERS30 UNION ALL
SELECT * FROM ORDERS31 UNION ALL
SELECT * FROM ORDERS32 UNION ALL
SELECT * FROM ORDERS33 UNION ALL
SELECT * FROM ORDERS34 UNION ALL
SELECT * FROM ORDERS35 UNION ALL
SELECT * FROM ORDERS36 UNION ALL
SELECT * FROM ORDERS37 UNION ALL
SELECT * FROM ORDERS38 UNION ALL
SELECT * FROM ORDERS39 UNION ALL
SELECT * FROM ORDERS40 UNION ALL
SELECT * FROM ORDERS41 UNION ALL
SELECT * FROM ORDERS42 UNION ALL
SELECT * FROM ORDERS43 UNION ALL
SELECT * FROM ORDERS44 UNION ALL
SELECT * FROM ORDERS45 UNION ALL
SELECT * FROM ORDERS46 UNION ALL
SELECT * FROM ORDERS47 UNION ALL
SELECT * FROM ORDERS48 UNION ALL
SELECT * FROM ORDERS49 UNION ALL
SELECT * FROM ORDERS50 UNION ALL
SELECT * FROM ORDERS51 UNION ALL
SELECT * FROM ORDERS52 UNION ALL
SELECT * FROM ORDERS53 UNION ALL
SELECT * FROM ORDERS54 UNION ALL
SELECT * FROM ORDERS55 UNION ALL
SELECT * FROM ORDERS56 UNION ALL
SELECT * FROM ORDERS57 UNION ALL
SELECT * FROM ORDERS58 UNION ALL
SELECT * FROM ORDERS59 UNION ALL
SELECT * FROM ORDERS60 UNION ALL
SELECT * FROM ORDERS61 UNION ALL
SELECT * FROM ORDERS62 UNION ALL
SELECT * FROM ORDERS63 UNION ALL
SELECT * FROM ORDERS64 UNION ALL
SELECT * FROM ORDERS65 UNION ALL
SELECT * FROM ORDERS66 UNION ALL
SELECT * FROM ORDERS67 UNION ALL
SELECT * FROM ORDERS68 UNION ALL
SELECT * FROM ORDERS69 UNION ALL
SELECT * FROM ORDERS70 UNION ALL
SELECT * FROM ORDERS71 UNION ALL
SELECT * FROM ORDERS72 UNION ALL
SELECT * FROM ORDERS73 UNION ALL
SELECT * FROM ORDERS74 UNION ALL
SELECT * FROM ORDERS75 UNION ALL
SELECT * FROM ORDERS76 UNION ALL
SELECT * FROM ORDERS77 UNION ALL
SELECT * FROM ORDERS78 UNION ALL
SELECT * FROM ORDERS79 UNION ALL
SELECT * FROM ORDERS80 UNION ALL
SELECT * FROM ORDERS81 UNION ALL
SELECT * FROM ORDERS82 UNION ALL
SELECT * FROM ORDERS83 UNION ALL
SELECT * FROM ORDERS84 UNION ALL

```

```

SELECT * FROM ORDERS85 UNION ALL
SELECT * FROM ORDERS86 UNION ALL
SELECT * FROM ORDERS87 UNION ALL
SELECT * FROM ORDERS88 UNION ALL
SELECT * FROM ORDERS89 UNION ALL
SELECT * FROM ORDERS90 UNION ALL
SELECT * FROM ORDERS91 UNION ALL
SELECT * FROM ORDERS92 UNION ALL
SELECT * FROM ORDERS93 UNION ALL
SELECT * FROM ORDERS94 UNION ALL
SELECT * FROM ORDERS95 UNION ALL
SELECT * FROM ORDERS96 UNION ALL
SELECT * FROM ORDERS97 UNION ALL
SELECT * FROM ORDERS98 UNION ALL
SELECT * FROM ORDERS99 UNION ALL
SELECT * FROM ORDERS100 UNION ALL
SELECT * FROM ORDERS101 UNION ALL
SELECT * FROM ORDERS102 UNION ALL
SELECT * FROM ORDERS103 UNION ALL
SELECT * FROM ORDERS104 UNION ALL
SELECT * FROM ORDERS105 UNION ALL
SELECT * FROM ORDERS106 UNION ALL
SELECT * FROM ORDERS107 UNION ALL
SELECT * FROM ORDERS108 UNION ALL
SELECT * FROM ORDERS109 UNION ALL
SELECT * FROM ORDERS110 UNION ALL
SELECT * FROM ORDERS111 UNION ALL
SELECT * FROM ORDERS112 UNION ALL
SELECT * FROM ORDERS113 UNION ALL
SELECT * FROM ORDERS114 UNION ALL
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 UNION ALL
SELECT * FROM ORDERS121 UNION ALL
SELECT * FROM ORDERS122 UNION ALL
SELECT * FROM ORDERS123 UNION ALL
SELECT * FROM ORDERS124 UNION ALL
SELECT * FROM ORDERS125 UNION ALL
SELECT * FROM ORDERS126 UNION ALL
SELECT * FROM ORDERS127 UNION ALL
SELECT * FROM ORDERS128 UNION ALL
SELECT * FROM ORDERS129 UNION ALL
SELECT * FROM ORDERS130 UNION ALL
SELECT * FROM ORDERS131 UNION ALL
SELECT * FROM ORDERS132 UNION ALL
SELECT * FROM ORDERS133 UNION ALL
SELECT * FROM ORDERS134 UNION ALL
SELECT * FROM ORDERS135 UNION ALL
SELECT * FROM ORDERS136 UNION ALL
SELECT * FROM ORDERS137 UNION ALL
SELECT * FROM ORDERS138 UNION ALL
SELECT * FROM ORDERS139 UNION ALL
SELECT * FROM ORDERS140 UNION ALL
SELECT * FROM ORDERS141 UNION ALL
SELECT * FROM ORDERS142 UNION ALL
SELECT * FROM ORDERS143 UNION ALL
SELECT * FROM ORDERS144 UNION ALL
SELECT * FROM ORDERS145 UNION ALL
SELECT * FROM ORDERS146 UNION ALL

```



```

SELECT * FROM ORDER_LINE153 UNION ALL
SELECT * FROM ORDER_LINE154 UNION ALL
SELECT * FROM ORDER_LINE155 UNION ALL
SELECT * FROM ORDER_LINE156 UNION ALL
SELECT * FROM ORDER_LINE157 UNION ALL
SELECT * FROM ORDER_LINE158 UNION ALL
SELECT * FROM ORDER_LINE159 UNION ALL
SELECT * FROM ORDER_LINE160
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRWV STOCK.ddl

connect to TPCC in share mode;

DROP VIEW STOCK;

CREATE VIEW STOCK

```

(S_REMOTE_CNT,
S_QUANTITY,
S_ORDER_CNT,
S_YTD,
S_DATA,
S_DIST_01,
S_DIST_02,
S_DIST_03,
S_DIST_04,
S_DIST_05,
S_DIST_06,
S_DIST_07,
S_DIST_08,
S_DIST_09,
S_DIST_10,
S_I_ID,
S_W_ID

```

) AS SELECT * FROM STOCK1 UNION ALL

```

SELECT * FROM STOCK2 UNION ALL
SELECT * FROM STOCK3 UNION ALL
SELECT * FROM STOCK4 UNION ALL
SELECT * FROM STOCK5 UNION ALL
SELECT * FROM STOCK6 UNION ALL
SELECT * FROM STOCK7 UNION ALL
SELECT * FROM STOCK8 UNION ALL
SELECT * FROM STOCK9 UNION ALL
SELECT * FROM STOCK10 UNION ALL
SELECT * FROM STOCK11 UNION ALL
SELECT * FROM STOCK12 UNION ALL
SELECT * FROM STOCK13 UNION ALL
SELECT * FROM STOCK14 UNION ALL
SELECT * FROM STOCK15 UNION ALL
SELECT * FROM STOCK16 UNION ALL
SELECT * FROM STOCK17 UNION ALL
SELECT * FROM STOCK18 UNION ALL
SELECT * FROM STOCK19 UNION ALL
SELECT * FROM STOCK20 UNION ALL
SELECT * FROM STOCK21 UNION ALL
SELECT * FROM STOCK22 UNION ALL
SELECT * FROM STOCK23 UNION ALL
SELECT * FROM STOCK24 UNION ALL
SELECT * FROM STOCK25 UNION ALL
SELECT * FROM STOCK26 UNION ALL
SELECT * FROM STOCK27 UNION ALL

```

```

SELECT * FROM STOCK28 UNION ALL
SELECT * FROM STOCK29 UNION ALL
SELECT * FROM STOCK30 UNION ALL
SELECT * FROM STOCK31 UNION ALL
SELECT * FROM STOCK32 UNION ALL
SELECT * FROM STOCK33 UNION ALL
SELECT * FROM STOCK34 UNION ALL
SELECT * FROM STOCK35 UNION ALL
SELECT * FROM STOCK36 UNION ALL
SELECT * FROM STOCK37 UNION ALL
SELECT * FROM STOCK38 UNION ALL
SELECT * FROM STOCK39 UNION ALL
SELECT * FROM STOCK40 UNION ALL
SELECT * FROM STOCK41 UNION ALL
SELECT * FROM STOCK42 UNION ALL
SELECT * FROM STOCK43 UNION ALL
SELECT * FROM STOCK44 UNION ALL
SELECT * FROM STOCK45 UNION ALL
SELECT * FROM STOCK46 UNION ALL
SELECT * FROM STOCK47 UNION ALL
SELECT * FROM STOCK48 UNION ALL
SELECT * FROM STOCK49 UNION ALL
SELECT * FROM STOCK50 UNION ALL
SELECT * FROM STOCK51 UNION ALL
SELECT * FROM STOCK52 UNION ALL
SELECT * FROM STOCK53 UNION ALL
SELECT * FROM STOCK54 UNION ALL
SELECT * FROM STOCK55 UNION ALL
SELECT * FROM STOCK56 UNION ALL
SELECT * FROM STOCK57 UNION ALL
SELECT * FROM STOCK58 UNION ALL
SELECT * FROM STOCK59 UNION ALL
SELECT * FROM STOCK60 UNION ALL
SELECT * FROM STOCK61 UNION ALL
SELECT * FROM STOCK62 UNION ALL
SELECT * FROM STOCK63 UNION ALL
SELECT * FROM STOCK64 UNION ALL
SELECT * FROM STOCK65 UNION ALL
SELECT * FROM STOCK66 UNION ALL
SELECT * FROM STOCK67 UNION ALL
SELECT * FROM STOCK68 UNION ALL
SELECT * FROM STOCK69 UNION ALL
SELECT * FROM STOCK70 UNION ALL
SELECT * FROM STOCK71 UNION ALL
SELECT * FROM STOCK72 UNION ALL
SELECT * FROM STOCK73 UNION ALL
SELECT * FROM STOCK74 UNION ALL
SELECT * FROM STOCK75 UNION ALL
SELECT * FROM STOCK76 UNION ALL
SELECT * FROM STOCK77 UNION ALL
SELECT * FROM STOCK78 UNION ALL
SELECT * FROM STOCK79 UNION ALL
SELECT * FROM STOCK80 UNION ALL
SELECT * FROM STOCK81 UNION ALL
SELECT * FROM STOCK82 UNION ALL
SELECT * FROM STOCK83 UNION ALL
SELECT * FROM STOCK84 UNION ALL
SELECT * FROM STOCK85 UNION ALL
SELECT * FROM STOCK86 UNION ALL
SELECT * FROM STOCK87 UNION ALL
SELECT * FROM STOCK88 UNION ALL
SELECT * FROM STOCK89 UNION ALL

```

```

SELECT * FROM STOCK90 UNION ALL
SELECT * FROM STOCK91 UNION ALL
SELECT * FROM STOCK92 UNION ALL
SELECT * FROM STOCK93 UNION ALL
SELECT * FROM STOCK94 UNION ALL
SELECT * FROM STOCK95 UNION ALL
SELECT * FROM STOCK96 UNION ALL
SELECT * FROM STOCK97 UNION ALL
SELECT * FROM STOCK98 UNION ALL
SELECT * FROM STOCK99 UNION ALL
SELECT * FROM STOCK100 UNION ALL
SELECT * FROM STOCK101 UNION ALL
SELECT * FROM STOCK102 UNION ALL
SELECT * FROM STOCK103 UNION ALL
SELECT * FROM STOCK104 UNION ALL
SELECT * FROM STOCK105 UNION ALL
SELECT * FROM STOCK106 UNION ALL
SELECT * FROM STOCK107 UNION ALL
SELECT * FROM STOCK108 UNION ALL
SELECT * FROM STOCK109 UNION ALL
SELECT * FROM STOCK110 UNION ALL
SELECT * FROM STOCK111 UNION ALL
SELECT * FROM STOCK112 UNION ALL
SELECT * FROM STOCK113 UNION ALL
SELECT * FROM STOCK114 UNION ALL
SELECT * FROM STOCK115 UNION ALL
SELECT * FROM STOCK116 UNION ALL
SELECT * FROM STOCK117 UNION ALL
SELECT * FROM STOCK118 UNION ALL
SELECT * FROM STOCK119 UNION ALL
SELECT * FROM STOCK120 UNION ALL
SELECT * FROM STOCK121 UNION ALL
SELECT * FROM STOCK122 UNION ALL
SELECT * FROM STOCK123 UNION ALL
SELECT * FROM STOCK124 UNION ALL
SELECT * FROM STOCK125 UNION ALL
SELECT * FROM STOCK126 UNION ALL
SELECT * FROM STOCK127 UNION ALL
SELECT * FROM STOCK128 UNION ALL
SELECT * FROM STOCK129 UNION ALL
SELECT * FROM STOCK130 UNION ALL
SELECT * FROM STOCK131 UNION ALL
SELECT * FROM STOCK132 UNION ALL
SELECT * FROM STOCK133 UNION ALL
SELECT * FROM STOCK134 UNION ALL
SELECT * FROM STOCK135 UNION ALL
SELECT * FROM STOCK136 UNION ALL
SELECT * FROM STOCK137 UNION ALL
SELECT * FROM STOCK138 UNION ALL
SELECT * FROM STOCK139 UNION ALL
SELECT * FROM STOCK140 UNION ALL
SELECT * FROM STOCK141 UNION ALL
SELECT * FROM STOCK142 UNION ALL
SELECT * FROM STOCK143 UNION ALL
SELECT * FROM STOCK144 UNION ALL
SELECT * FROM STOCK145 UNION ALL
SELECT * FROM STOCK146 UNION ALL
SELECT * FROM STOCK147 UNION ALL
SELECT * FROM STOCK148 UNION ALL
SELECT * FROM STOCK149 UNION ALL
SELECT * FROM STOCK150 UNION ALL
SELECT * FROM STOCK151 UNION ALL

```

```

SELECT * FROM STOCK152 UNION ALL
SELECT * FROM STOCK153 UNION ALL
SELECT * FROM STOCK154 UNION ALL
SELECT * FROM STOCK155 UNION ALL
SELECT * FROM STOCK156 UNION ALL
SELECT * FROM STOCK157 UNION ALL
SELECT * FROM STOCK158 UNION ALL
SELECT * FROM STOCK159 UNION ALL
SELECT * FROM STOCK160
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW WAREHOUSE.ddl

```

connect to TPCC in share mode;
DROP VIEW WAREHOUSE;
CREATE VIEW WAREHOUSE
(W_NAME,
 W_STREET_1,
 W_STREET_2,
 W_CITY,
 W_STATE,
 W_ZIP,
 W_TAX,
 W_YTD,
 W_ID
 ) AS SELECT * FROM WAREHOUSE1 UNION ALL
SELECT * FROM WAREHOUSE2 UNION ALL
SELECT * FROM WAREHOUSE3 UNION ALL
SELECT * FROM WAREHOUSE4 UNION ALL
SELECT * FROM WAREHOUSE5 UNION ALL
SELECT * FROM WAREHOUSE6 UNION ALL
SELECT * FROM WAREHOUSE7 UNION ALL
SELECT * FROM WAREHOUSE8
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;;

```

GEN CUSTOMER.sh

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 400 -f1
/flats/F1_001/customer_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 401 800 -f1
/flats/F1_001/customer_001_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 801 1200 -f1
/flats/F1_001/customer_001_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1201 1600 -f1
/flats/F1_001/customer_001_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1601 2000 -f1
/flats/F1_002/customer_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2001 2400 -f1
/flats/F1_002/customer_002_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2401 2800 -f1
/flats/F1_002/customer_002_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2801 3200 -f1
/flats/F1_002/customer_002_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3201 3600 -f1
/flats/F1_003/customer_003_1.dat

```

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3601 4000 -f1
/flats/F1_003/customer_003_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4001 4400 -f1
/flats/F1_003/customer_003_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4401 4800 -f1
/flats/F1_003/customer_003_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4801 5200 -f1
/flats/F1_004/customer_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5201 5600 -f1
/flats/F1_004/customer_004_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5601 6000 -f1
/flats/F1_004/customer_004_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6001 6400 -f1
/flats/F1_004/customer_004_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6401 6800 -f1
/flats/F1_005/customer_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6801 7200 -f1
/flats/F1_005/customer_005_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7201 7600 -f1
/flats/F1_005/customer_005_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7601 8000 -f1
/flats/F1_005/customer_005_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8001 8400 -f1
/flats/F1_006/customer_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8401 8800 -f1
/flats/F1_006/customer_006_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8801 9200 -f1
/flats/F1_006/customer_006_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9201 9600 -f1
/flats/F1_006/customer_006_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9601 10000 -f1
/flats/F1_007/customer_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10001 10400 -f1
/flats/F1_007/customer_007_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10401 10800 -f1
/flats/F1_007/customer_007_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10801 11200 -f1
/flats/F1_007/customer_007_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11201 11600 -f1
/flats/F1_008/customer_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11601 12000 -f1
/flats/F1_008/customer_008_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12001 12400 -f1
/flats/F1_008/customer_008_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12401 12800 -f1
/flats/F1_008/customer_008_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12801 13200 -f1
/flats/F1_009/customer_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13201 13600 -f1
/flats/F1_009/customer_009_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13601 14000 -f1
/flats/F1_009/customer_009_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14001 14400 -f1
/flats/F1_009/customer_009_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14401 14800 -f1
/flats/F1_010/customer_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14801 15200 -f1
/flats/F1_010/customer_010_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15201 15600 -f1
/flats/F1_010/customer_010_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15601 16000 -f1
/flats/F1_010/customer_010_4.dat

```

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16001 16400 -f1
/flats/F1_011/customer_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16401 16800 -f1
/flats/F1_011/customer_011_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16801 17200 -f1
/flats/F1_011/customer_011_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17201 17600 -f1
/flats/F1_011/customer_011_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17601 18000 -f1
/flats/F1_012/customer_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18001 18400 -f1
/flats/F1_012/customer_012_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18401 18800 -f1
/flats/F1_012/customer_012_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18801 19200 -f1
/flats/F1_012/customer_012_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19201 19600 -f1
/flats/F1_013/customer_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19601 20000 -f1
/flats/F1_013/customer_013_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20001 20400 -f1
/flats/F1_013/customer_013_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20401 20800 -f1
/flats/F1_013/customer_013_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20801 21200 -f1
/flats/F1_014/customer_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21201 21600 -f1
/flats/F1_014/customer_014_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21601 22000 -f1
/flats/F1_014/customer_014_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22001 22400 -f1
/flats/F1_014/customer_014_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22401 22800 -f1
/flats/F1_015/customer_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22801 23200 -f1
/flats/F1_015/customer_015_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23201 23600 -f1
/flats/F1_015/customer_015_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23601 24000 -f1
/flats/F1_015/customer_015_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24001 24400 -f1
/flats/F1_016/customer_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24401 24800 -f1
/flats/F1_016/customer_016_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24801 25200 -f1
/flats/F1_016/customer_016_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25201 25600 -f1
/flats/F1_016/customer_016_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25601 26000 -f1
/flats/F1_017/customer_017_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26001 26400 -f1
/flats/F1_017/customer_017_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26401 26800 -f1
/flats/F1_017/customer_017_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26801 27200 -f1
/flats/F1_017/customer_017_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27201 27600 -f1
/flats/F1_018/customer_018_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27601 28000 -f1
/flats/F1_018/customer_018_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28001 28400 -f1
/flats/F1_018/customer_018_3.dat

```


/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 251601 252000 -f1
/flats/F1_158/customer_158_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 252001 252400 -f1
/flats/F1_158/customer_158_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 252401 252800 -f1
/flats/F1_158/customer_158_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 252801 253200 -f1
/flats/F1_159/customer_159_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 253201 253600 -f1
/flats/F1_159/customer_159_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 253601 254000 -f1
/flats/F1_159/customer_159_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 254001 254400 -f1
/flats/F1_159/customer_159_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 254401 254800 -f1
/flats/F1_160/customer_160_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 254801 255200 -f1
/flats/F1_160/customer_160_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 255201 255600 -f1
/flats/F1_160/customer_160_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 255601 256000 -f1
/flats/F1_160/customer_160_4.dat

GEN DISTRICT.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1 32000 -f1
/flats/F1_001/district_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 32001 64000 -f1
/flats/F1_002/district_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 64001 96000 -f1
/flats/F1_003/district_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 96001 128000 -f1
/flats/F1_004/district_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 128001 160000 -f1
/flats/F1_005/district_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 160001 192000 -f1
/flats/F1_006/district_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 192001 224000 -f1
/flats/F1_007/district_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 224001 256000 -f1
/flats/F1_008/district_008_1.dat

GEN HISTORY.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1 4000 -f1
/flats/F1_001/history_001_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 4001 8000 -f1
/flats/F1_001/history_001_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 8001 12000 -f1
/flats/F1_001/history_001_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 12001 16000 -f1
/flats/F1_001/history_001_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 16001 20000 -f1
/flats/F1_002/history_002_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 20001 24000 -f1
/flats/F1_002/history_002_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 24001 28000 -f1
/flats/F1_002/history_002_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 28001 32000 -f1
/flats/F1_002/history_002_4.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 32001 36000 -f1
/flats/F1_003/history_003_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 36001 40000 -f1
/flats/F1_003/history_003_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 40001 44000 -f1
/flats/F1_003/history_003_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 44001 48000 -f1
/flats/F1_003/history_003_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 48001 52000 -f1
/flats/F1_004/history_004_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 52001 56000 -f1
/flats/F1_004/history_004_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 56001 60000 -f1
/flats/F1_004/history_004_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 60001 64000 -f1
/flats/F1_004/history_004_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 64001 68000 -f1
/flats/F1_005/history_005_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 68001 72000 -f1
/flats/F1_005/history_005_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 72001 76000 -f1
/flats/F1_005/history_005_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 76001 80000 -f1
/flats/F1_005/history_005_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 80001 84000 -f1
/flats/F1_006/history_006_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 84001 88000 -f1
/flats/F1_006/history_006_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 88001 92000 -f1
/flats/F1_006/history_006_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 92001 96000 -f1
/flats/F1_006/history_006_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 96001 100000 -f1
/flats/F1_007/history_007_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 100001 104000 -f1
/flats/F1_007/history_007_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 104001 108000 -f1
/flats/F1_007/history_007_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 108001 112000 -f1
/flats/F1_007/history_007_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 112001 116000 -f1
/flats/F1_008/history_008_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 116001 120000 -f1
/flats/F1_008/history_008_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 120001 124000 -f1
/flats/F1_008/history_008_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 124001 128000 -f1
/flats/F1_008/history_008_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 128001 132000 -f1
/flats/F1_009/history_009_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 132001 136000 -f1
/flats/F1_009/history_009_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 136001 140000 -f1
/flats/F1_009/history_009_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 140001 144000 -f1
/flats/F1_009/history_009_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 144001 148000 -f1
/flats/F1_010/history_010_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 148001 152000 -f1
/flats/F1_010/history_010_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 152001 156000 -f1
/flats/F1_010/history_010_3.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 156001 160000 -f1
/flats/F1_010/history_010_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 160001 164000 -f1
/flats/F1_011/history_011_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 164001 168000 -f1
/flats/F1_011/history_011_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 168001 172000 -f1
/flats/F1_011/history_011_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 172001 176000 -f1
/flats/F1_011/history_011_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 176001 180000 -f1
/flats/F1_012/history_012_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 180001 184000 -f1
/flats/F1_012/history_012_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 184001 188000 -f1
/flats/F1_012/history_012_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 188001 192000 -f1
/flats/F1_012/history_012_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 192001 196000 -f1
/flats/F1_013/history_013_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 196001 200000 -f1
/flats/F1_013/history_013_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 200001 204000 -f1
/flats/F1_013/history_013_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 204001 208000 -f1
/flats/F1_013/history_013_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 208001 212000 -f1
/flats/F1_014/history_014_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 212001 216000 -f1
/flats/F1_014/history_014_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 216001 220000 -f1
/flats/F1_014/history_014_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 220001 224000 -f1
/flats/F1_014/history_014_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 224001 228000 -f1
/flats/F1_015/history_015_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 228001 232000 -f1
/flats/F1_015/history_015_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 232001 236000 -f1
/flats/F1_015/history_015_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 236001 240000 -f1
/flats/F1_015/history_015_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 240001 244000 -f1
/flats/F1_016/history_016_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 244001 248000 -f1
/flats/F1_016/history_016_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 248001 252000 -f1
/flats/F1_016/history_016_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 252001 256000 -f1
/flats/F1_016/history_016_4.dat

GEN ITEM.sh

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 5 -f1
/flats/F1_01/item_1.dat

GEN NEW ORDER.sh


```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER159 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_159/customer_159_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER159;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER159 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_159/customer_159_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER159;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER159 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_159/customer_159_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER159;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/customer_160_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER160;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/customer_160_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER160;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/customer_160_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER160;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/customer_160_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 12000000 INSERT INTO CUSTOMER160;
COMMIT WORK;

```

```
CONNECT RESET;
```

LOAD DISTRICT ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_001/district_001_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_002/district_002_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_003/district_003_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_004/district_004_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_005/district_005_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_006/district_006_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_007/district_007_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_008/district_008_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
DISTRICT8;
COMMIT WORK;
CONNECT RESET;

```

LOAD HISTORY ALL.ddl

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

```

LOAD FROM /flats/F1_01/history_01_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY1 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_02/history_02_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY2 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_03/history_03_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY3 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_04/history_04_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY4 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_05/history_05_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY5 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_06/history_06_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY6 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_07/history_07_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY7 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/F1_08/history_08_1.dat OF DEL MODIFIED
BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO
HISTORY8 NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;

```

LOAD ITEM 1.ddl

```
CONNECT TO TPCC IN SHARE MODE;
```

```

IMPORT FROM /flats/F1_01/item_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT
1000 INSERT INTO ITEM;
COMMIT WORK;

```



```

UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK158 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_158/stock_158_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK158;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK159 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_159/stock_159_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK159;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK159 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_159/stock_159_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK159;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK159 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_159/stock_159_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK159;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK159 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_159/stock_159_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK159;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/stock_160_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK160;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/stock_160_2.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK160;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/stock_160_3.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK160;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE STOCK160 ACTIVATE NOT LOGGED
INITIALLY;
IMPORT FROM /flats/F1_160/stock_160_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS COMPOUND=50
COMMITCOUNT 40000000 INSERT INTO STOCK160;
COMMIT WORK;
CONNECT RESET;

```

LOAD WAREHOUSE ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_001/warehouse_001_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_002/warehouse_002_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_003/warehouse_003_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_004/warehouse_004_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_005/warehouse_005_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_006/warehouse_006_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS

```

```

ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_007/warehouse_007_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/F1_008/warehouse_008_1.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS USEDEFAULTS
ALLOW WRITE ACCESS COMMITCOUNT 1000 INSERT INTO
WAREHOUSE8;
COMMIT WORK;
CONNECT RESET;

```

RNST CUSTOMER.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER1 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER2 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER3 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER4 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER5 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER6 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER7 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER8 AND INDEXES
ALL;
COMMIT WORK;

```


connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER157 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER158 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER159 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER160 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_DISTRICT.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.DISTRICT8 AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_HISTORY.ddl

connect to TPCC in share mode;

RUNSTATS ON TABLE tpcc.HISTORY1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.HISTORY16 AND INDEXES ALL;
COMMIT WORK;

connect reset;

RNST_ITEM.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ITEM AND INDEXES ALL;
COMMIT WORK;
connect reset;

RNST_NEW_ORDER.ddl

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERA8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.NEW_ORDERB2 AND INDEXES ALL;
COMMIT WORK;
connect reset;


```

connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK141 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK142 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK143 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK144 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK145 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK146 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK147 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK148 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK149 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK150 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK151 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK152 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK153 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK154 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK155 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

```

```

RUNSTATS ON TABLE tpcc.STOCK156 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK157 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK158 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK159 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.STOCK160 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

RNST WAREHOUSE.ddl

```

connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE1 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE2 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE3 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE4 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE5 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE6 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE7 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE8 AND INDEXES
ALL;

```

```

COMMIT WORK;
connect reset;

```

SORT ORDERS.sh

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_001/orders_001_1.dat >
/flats/F1_001/orders_001_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_001/orders_001_1.dat.sorted
/flats/F1_001/orders_001_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_001/orders_001_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_001/orders_001_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_001/orders_001_2.dat >
/flats/F1_001/orders_001_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_001/orders_001_2.dat.sorted
/flats/F1_001/orders_001_2.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_001/orders_001_2.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_001/orders_001_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_001/orders_001_3.dat >
/flats/F1_001/orders_001_3.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_001/orders_001_3.dat.sorted
/flats/F1_001/orders_001_3.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_001/orders_001_3.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_001/orders_001_3.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_001/orders_001_4.dat >
/flats/F1_001/orders_001_4.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_001/orders_001_4.dat.sorted
/flats/F1_001/orders_001_4.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_001/orders_001_4.dat FAILED."

```



```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_011/orders_011_1.dat >
/flats/F1_011/orders_011_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_011/orders_011_1.dat.sorted
/flats/F1_011/orders_011_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_011/orders_011_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_011/orders_011_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_012/orders_012_1.dat >
/flats/F1_012/orders_012_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_012/orders_012_1.dat.sorted
/flats/F1_012/orders_012_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_012/orders_012_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_012/orders_012_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_013/orders_013_1.dat >
/flats/F1_013/orders_013_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_013/orders_013_1.dat.sorted
/flats/F1_013/orders_013_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_013/orders_013_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_013/orders_013_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```



```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_053/orders_053_1.dat >
/flats/F1_053/orders_053_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_053/orders_053_1.dat.sorted
/flats/F1_053/orders_053_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_053/orders_053_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_053/orders_053_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_054/orders_054_1.dat >
/flats/F1_054/orders_054_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_054/orders_054_1.dat.sorted
/flats/F1_054/orders_054_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_054/orders_054_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_054/orders_054_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_055/orders_055_1.dat >
/flats/F1_055/orders_055_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_055/orders_055_1.dat.sorted
/flats/F1_055/orders_055_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_055/orders_055_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_055/orders_055_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```



```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_098/orders_098_1.dat >
/flats/F1_098/orders_098_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_098/orders_098_1.dat.sorted
/flats/F1_098/orders_098_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_098/orders_098_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_098/orders_098_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_099/orders_099_1.dat >
/flats/F1_099/orders_099_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_099/orders_099_1.dat.sorted
/flats/F1_099/orders_099_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_099/orders_099_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_099/orders_099_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```

```

sort -t'|' -T /ixtemp/sort +5n +6n +7n
/flats/F1_100/orders_100_1.dat >
/flats/F1_100/orders_100_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_100/orders_100_1.dat.sorted
/flats/F1_100/orders_100_1.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_100/orders_100_1.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_100/orders_100_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."

```



```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_9.dat >
/flats/F1_008/neworder_008_9.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_9.dat.sorted
/flats/F1_008/neworder_008_9.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_9.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_9.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_10.dat >
/flats/F1_008/neworder_008_10.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_10.dat.sorted
/flats/F1_008/neworder_008_10.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_10.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_10.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_11.dat >
/flats/F1_008/neworder_008_11.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_11.dat.sorted
/flats/F1_008/neworder_008_11.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_11.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_11.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_12.dat >
/flats/F1_008/neworder_008_12.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_12.dat.sorted
/flats/F1_008/neworder_008_12.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_12.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_12.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

```

sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_13.dat >
/flats/F1_008/neworder_008_13.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_13.dat.sorted
/flats/F1_008/neworder_008_13.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_13.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_13.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_14.dat >
/flats/F1_008/neworder_008_14.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_14.dat.sorted
/flats/F1_008/neworder_008_14.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_14.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_14.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_15.dat >
/flats/F1_008/neworder_008_15.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_15.dat.sorted
/flats/F1_008/neworder_008_15.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_15.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_15.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi
sort -t'|' -T /ixtemp/sort +2n +1n +0n
/flats/F1_008/neworder_008_16.dat >
/flats/F1_008/neworder_008_16.dat.sorted
if [ $? = 0 ]
then
mv /flats/F1_008/neworder_008_16.dat.sorted
/flats/F1_008/neworder_008_16.dat
else
echo "**** WARNING ****"
echo "Sort of /flats/F1_008/neworder_008_16.dat FAILED."
echo "Please ensure that the source file
(/flats/F1_008/neworder_008_16.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp/sort)."
fi

```

C.2 Data Generation Code

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 - 2004
## 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 -
IS$(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)

```



```

    fprintf(stderr, "gendata: Don't understand argument:
%s\n", argv[i]);
    exit(-1);
}
}
/* ***** */
/* Validate Command Line Arguments */
/* ***** */
/* Validate Table Argument */
if (option < 3 || option > 11 || option == 10)
{
    fprintf(stderr, "gendata: Invalid table selected: %d\n", option);
    exit(-1);
}
/* Validate Delimiter Argument */
if (delim == NULL) {
    // default delimiter is used for IMPORT & LOAD, no changes
    necessary
    using_rctload = 0;
} else if (strlen(delim) == 1 && !isalnum(delim[0]) &&
    delim[0] != '.' && delim[0] != '%')
{
    // user-supplied delimiter used for rctload
    InitFormatStrings(delim[0]);
    using_rctload = 1;
} else {
    fprintf(stderr, "gendata: Invalid delimiter specified:
%s\n", delim);
    exit(-1);
}
/* Validate File/Pipe Arguments */
if (option != 9 && outtype1 > 0 && outtype2 > 0)
{
    fprintf(stderr, "gendata: Specifying two output file/pipes
allowed only when generating\norders/orderline.\n");
    exit(-1);
}
if (option == 9 && ((outtype1 == 0) || (outtype2 == 0)))
{
    fprintf(stderr, "gendata: Must specify two output file/pipes
when generating orders/orderline.\n");
    exit(-1);
}
if (outtype1 == 0 || outname1 == NULL || strcmp(outname1, "")
== 0)
{
    fprintf(stderr, "gendata: Invalid 1st output file/pipe
specified.\n");
    exit(-1);
}
if (option == 9 && (outtype2 == 0 || outname2 == NULL ||
strcmp(outname2, "") == 0))
{
    fprintf(stderr, "gendata: Invalid 2nd output file/pipe
specified.\n");
    exit(-1);
}
}
/* Ensure O/OL flat files are opened in append mode. This is
required */
/* because we generate O/OL concurrently. See comments in
genload.pl */
/* for further details on why this is necessary. */

```

```

if (option == 9)
{
    if (outtype1 == IOH_FILE) outtype1 = IOH_FILE_APPEND;
    if (outtype2 == IOH_FILE) outtype2 = IOH_FILE_APPEND;
}
/* Validate Range Arguments */
if (ware_start <= 0 || ware_start > WAREHOUSES) {
    fprintf(stderr, "gendata: Invalid range starting value:
%d\n", ware_start);
    exit(-1);
}
if (ware_end <= 0 || ware_end > WAREHOUSES || ware_end
< ware_start) {
    fprintf(stderr, "gendata: Invalid range ending value:
%d\n", ware_end);
    exit(-1);
}
/* ***** */
/* 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] ;
    sqlint32 item_price ;

```

```

char item_data[51] ;
IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];
initialize_random(13,42);
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_integer( 100, 10000 ) ;
    /* 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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nITEM table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nITEM table FAILED at (l %d) after %8.2f
seconds.\n\n", item_num, elapsed);
    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];
initialize_random(7,11);
timestamp1 = current_time();
rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto stock_done; }
for (stock_num = 1; stock_num <=
STOCK_PER_WAREHOUSE; stock_num++)
{
    if (!quiet_mode && (stock_num%500 == 0))
    {
        fprintf(stdout, "STOCK for Item #%d\n", stock_num);
        fflush(stdout);
    }
    for (ware_num = ware_start; ware_num <= ware_end;
ware_num++)
    {
        stock_quant = rand_integer( 10, 100 );
        create_random_a_string( stock_dist_01, 24, 24);
        create_random_a_string( stock_dist_02, 24, 24);
        create_random_a_string( stock_dist_03, 24, 24);
        create_random_a_string( stock_dist_04, 24, 24);
        create_random_a_string( stock_dist_05, 24, 24);
        create_random_a_string( stock_dist_06, 24, 24);
        create_random_a_string( stock_dist_07, 24, 24);
        create_random_a_string( stock_dist_08, 24, 24);
        create_random_a_string( stock_dist_09, 24, 24);
        create_random_a_string( stock_dist_10, 24, 24);
        /* create ORIGINAL field */
        create_a_string_with_original( stock_data, 26, 50, 10 );
        s_ytd = s_order_cnt = s_remote_cnt = 0;
        numBytes = sprintf(Buffer, fmtStock,
            s_remote_cnt,
            stock_quant,
            s_order_cnt,
            s_ytd,
            stock_data,
            stock_dist_01,
            stock_dist_02,
            stock_dist_03,
            stock_dist_04,
            stock_dist_05,
            stock_dist_06,
            stock_dist_07,
            stock_dist_08,
            stock_dist_09,
            stock_dist_10,
            stock_num,
            ware_num);
        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto stock_done; }
    } /* end for... */
} /* end for... */
rc = GenericClose(&hnd);
stock_done:
timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nSTOCK table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
}

```

```

} else {
    fprintf(stderr, "\nSTOCK table FAILED at (S %d W %d) after
%8.2f 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] ;
    sqlint32 ware_tax ;
    sqlint64 ware_YTD ;
    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];
    initialize_random(23,111);
    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))
        {
            //@dxxxxxmtmte
            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_integer(0, 2000);
        ware_YTD = 30000000;
        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) {
    fprintf(stdout, "\nWAREHOUSE table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after
%8.2f seconds.\n\n", ware_num, elapsed);
    fflush(stderr);
}
}
/*-----*/
/* generate dist table */
/*-----*/
void gen_dist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    char dist_name[11] ;
    char dist_street_1[21] ;
    char dist_street_2[21] ;
    char dist_city[21] ;
    char dist_state[3] ;
    char dist_zip[10] ;
    sqlint32 dist_tax ;
    sqlint32 next_o_id ;
    sqlint64 dist_YTD ;
    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];
    next_o_id = CUSTOMERS_PER_DISTRICT + 1;
    initialize_random(44,73);
    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_integer(0, 2000);
dist_YTD = 3000000;
numBytes = sprintf(Buffer, fmtDist,
    next_o_id,
    dist_tax,
    dist_YTD,
    dist_name,
    dist_street_1,
    dist_street_2,
    dist_city,
    dist_state,
    dist_zip,
    dist_num,
    ware_num);
rc = GenericWrite(&hnd, Buffer, numBytes);
if (rc != 0) { goto dist_done; }
} /* end for... */
} /* end for... */
rc = GenericClose(&hnd);
dist_done:
timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nDISTRICT table generated in %8.2f
seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nDISTRICT table FAILED at (W %d D %d)
after %8.2f seconds.\n\n", ware_num, dist_num, elapse);
    fflush(stderr);
}
}
/*-----*/
/* generate customer table */
/*-----*/
void gen_cust_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    char cust_last[17];
    char cust_middle[3];
    char cust_first[17];
    char cust_street_1[21];
    char cust_street_2[21];
    char cust_city[21];
    char cust_state[3];
    char cust_zip[10];
    char cust_phone[17];
    char cust_credit[3];
    char cust_data[501];
    sqlint32 cust_discount;
    sqlint64 currtmstp;
    sqlint64 cust_balance;
    sqlint64 cust_YTD_payment;
    sqlint64 cust_credit_lim;
    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];
    int len, pos;
    initialize_random(10,64);
    timestamp1 = current_time();

```

```

rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto cust_done; }
strcpy(cust_middle, "OE");
currtmstp = time(NULL);
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_integer(0, 5000);
            /* create customer data */
            create_random_a_string(cust_data, 300, 500);
            /* pad customer data (only for non-rcload) */
            if (using_rctload == 0) {
                for (pos=strlen(cust_data); pos<500; pos++)
                    cust_data[pos] = ' ';
                cust_data[500] = '\0';
            }
            cust_credit_lim = 5000000;
            cust_balance = -1000;
            cust_YTD_payment = 1000;
            numBytes = sprintf(Buffer, fmtCust,
                cust_num,
                cust_state,
                cust_zip,
                cust_phone,
                currtmstp,
                cust_credit_lim,
                cust_middle,

```

```

                cust_credit,
                cust_discount,
                cust_data,
                cust_last,
                cust_first,
                cust_street_1,
                cust_street_2,
                cust_city,
                dist_num,
                ware_num,
                0,
                cust_balance,
                cust_YTD_payment,
                1);
            rc = GenericWrite(&hnd, Buffer, numBytes);
            if (rc != 0) { goto cust_done; }
        } /* end for district... */
    } /* end for warehouse... */
} /* end for customer... */
rc = GenericClose(&hnd);
cust_done:
timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nCUSTOMER table generated in %8.2f
seconds.\n\n", elapse);
    fflush(stdout);
} else {
    fprintf(stderr, "\nCUSTOMER table FAILED at (W %d D %d)
C %d) after %8.2f seconds.\n\n", ware_num, dist_num,
cust_num, elapse);
    fflush(stderr);
}
}
/*-----*/
/* generate hist table */
/*-----*/
void gen_hist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    char hist_data[25];
    sqlint64 currtmstp;
    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];
    initialize_random(15,63);
    timestamp1 = current_time();
    rc = GenericOpen(&hnd, outtype1, outname1);
    if (rc != 0) { goto hist_done; }
    currtmstp = time(NULL);
    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,
        currtmstp,
        1000,
        hist_data);
    rc = GenericWrite(&hnd, Buffer, numBytes);
    if (rc != 0) { goto hist_done; }
  } /* end for customer... */
} /* end for district... */
} /* end for warehouse... */
rc = GenericClose(&hnd);
hist_done:
timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
  fprintf(stdout, "\nHISTORY table generated in %8.2f
seconds.\n\n", elapse);
  fflush(stdout);
} else {
  fprintf(stderr, "\nHISTORY table FAILED at (W %d D %d C
%d) after %8.2f seconds.\n\n", ware_num, dist_num, cust_num,
elapse);
  fflush(stderr);
}
}
/*-----*/
/* generate nu_ord table */
/*-----*/
void gen_nu_ord_tbl( void )
{
  sqlint32 ware_num = 0 ;
  sqlint32 dist_num = 0 ;
  sqlint32 nu_ord_id = 0 ;
  int nu_ord_hi ;
  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];
  /* compute maximum and minimum
order numbers for this
district */
  nu_ord_hi = CUSTOMERS_PER_DISTRICT -
NU_ORDERS_PER_DISTRICT + 1;
  if (nu_ord_hi < 0) {
    nu_ord_hi = CUSTOMERS_PER_DISTRICT -
(CUSTOMERS_PER_DISTRICT / 3) + 1;
    fprintf(stderr, "\n*** WARNING ****
NU_ORDERS_PER_DISTRICT is >
CUSTOMERS_PER_DISTRICT\n");
    fprintf(stderr, "          Check the values in file lval.h\n");
    fprintf(stderr, "          Loading New-Order with 1/3 of
CUSTOMERS_PER_DISTRICT\n");
  }
}

```

```

initialize_random(99,37);
timestamp1 = current_time();
rc = GenericOpen(&hnd, outtype1, outname1);
if (rc != 0) { goto neword_done; }
for (nu_ord_id = nu_ord_hi;
    nu_ord_id <= CUSTOMERS_PER_DISTRICT;
    nu_ord_id++)
  {
    if (!quiet_mode) {
      fprintf(stdout, "NEW_ORDER for Customer %d:\n",
nu_ord_id);
      fflush(stdout);
    }
    for (ware_num = ware_start; ware_num <= ware_end;
        ware_num++)
      {
        for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
          {
            numBytes = sprintf(Buffer, fmtNewOrd,
                nu_ord_id,
                dist_num,
                ware_num);
            rc = GenericWrite(&hnd, Buffer, numBytes);
            if (rc != 0) { goto neword_done; }
          } /* end for... */
        } /* end for... */
      rc = GenericClose(&hnd);
neword_done:
timestamp2 = current_time();
elapse = timestamp2 - timestamp1;
if (rc == 0) {
  fprintf(stdout, "\nNEW_ORDER table generated in %8.2f
seconds.\n\n", elapse);
  fflush(stdout);
} else {
  fprintf(stderr, "\nNEW_ORDER table FAILED at (W %d D %d
O %d) after %8.2f seconds.\n\n", ware_num, dist_num,
nu_ord_id, elapse);
  fflush(stderr);
}
}
/*-----*/
/* generate order and order_line tables */
/*-----*/
void gen_ordr_tbl( void )
{
  sqlint32 ware_num = 0 ;
  sqlint32 dist_num = 0 ;
  sqlint32 cust_num = 0 ;
  sqlint32 ord_num = 0 ;
  sqlint32 ordr_carrier_id;
  sqlint32 ordr_ol_cnt;
  sqlint32 oline_ol_num;
  sqlint32 oline_item_num;
  sqlint32 oline_amount;
  char oline_dist_info[25];
  sqlint64 nulltmstp = 0;
  sqlint64 currtmstp;
  IOH_NUM numBytes;
  ioHandle hnd1, hnd2;
}

```

```

char Buffer[1024];
oline_dist_info[24] = '\0';
initialize_random(42,13);
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; }
currtmstp = time(NULL);
for (ware_num = ware_start; ware_num <= ware_end;
    ware_num++)
  {
    if (!quiet_mode) {
      fprintf(stdout, "ORDERS & ORDER_LINE for Warehouse
%d\n", ware_num);
      fflush(stdout);
    }
    for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE; dist_num++)
      {
        if (!quiet_mode) {
          fprintf(stdout, "District %d\t", dist_num);
          fflush(stdout);
        }
        seed_1_3000();
        for (ord_num = 1; ord_num <=
CUSTOMERS_PER_DISTRICT; ord_num++)
          {
            if (ord_num < 2101)
              ordr_carrier_id = rand_integer( 1, 10 );
            else
              ordr_carrier_id = 0;
            cust_num = random_1_3000();
            ordr_ol_cnt =
rand_integer(MIN_OL_PER_ORDER, MAX_OL_PER_ORDER);
            numBytes = sprintf(Buffer, fmtOrdr,
                cust_num,
                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 :
rand_integer(1,999999)),
                    oline_item_num,
                    ware_num,
                    5,
                    oline_dist_info,
                    ord_num,

```



```

        dist_num,
        ware_num,
        oline_ol_num);
    rc2 = GenericWrite(&hnd2, Buffer, numBytes);
    if (rc2 != 0) { goto ool_done; }
} /* for order_line */
} /* for order */
} /* for dist */
} /* for ware */
rc1 = GenericClose(&hnd2);
rc2 = GenericClose(&hnd1);
ool_done:
    timestamp2 = current_time();
    elapse = timestamp2 - timestamp1;
    if (rc1 == 0 && rc2 == 0) {
        fprintf(stdout, "\nORDERS & ORDER_LINE table(s)
generated in %8.2f seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nORDERS & ORDER_LINE table(s) FAILED
at (W %d D %d O %d OL %d) after %8.2f
seconds.\n\n", ware_num, dist_num, ord_num, oline_ol_num,
elapse);
        fflush(stderr);
    }
}
// This routine will initialize the printf format strings and replace
the
// delimiter with the one provided. The pipe symbol is the
default.
void InitFormatStrings(char delim)
{
    char *p;
    // Check if Using Default Delimiter
    if (delim == '|') return;
    // Replace Delimiters
    while (p = strchr(fmtWare, '|')) { *p = delim; }
    while (p = strchr(fmtDist, '|')) { *p = delim; }
    while (p = strchr(fmtItem, '|')) { *p = delim; }
    while (p = strchr(fmtStock, '|')) { *p = delim; }
    while (p = strchr(fmtCust, '|')) { *p = delim; }
    while (p = strchr(fmtHist, '|')) { *p = delim; }
    while (p = strchr(fmtOrdr, '|')) { *p = delim; }
    while (p = strchr(fmtOLine, '|')) { *p = delim; }
    while (p = strchr(fmtNewOrd, '|')) { *p = delim; }
}
void ScalingReport(void)
{
    /* Print Scaling Values */
    fprintf(stdout, "Scaling Values in Use\n");
    fprintf(stdout, "-----\n");
    fprintf(stdout, "Warehouses:      %d\n", WAREHOUSES);
    fprintf(stdout, "Districts/Warehouse:  %d\n",
DISTRICTS_PER_WAREHOUSE);
    fprintf(stdout, "Customers/District:   %d\n",
CUSTOMERS_PER_DISTRICT);
    fprintf(stdout, "Items:                 %d\n", ITEMS);
    fprintf(stdout, "Stock/Warehouse:      %d\n",
STOCK_PER_WAREHOUSE);
    fprintf(stdout, "Min Order Lines/Order: %d\n",
MIN_OL_PER_ORDER);

```

```

    fprintf(stdout, "Max Order Lines/Order: %d\n",
MAX_OL_PER_ORDER);
    fprintf(stdout, "New Orders/District:  %d\n",
NU_ORDERS_PER_DISTRICT);
    fprintf(stdout, "-----\n");
}

```

dbgen/tpccrnd.c

```

/*****
****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp.
1996 - 2004
** 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 "db2tpcc.h"
#include "lval.h"
static char tbl_cust[CUSTOMERS_PER_DISTRICT];
static char alum[] =

"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMN
OPQRSTUVWXYZ";
static char *last_name_parts[] =
{
    "BAR",
    "OUGHT",
    "ABLE",
    "PRI",
    "PRES",
    "ESE",
    "ANTI",
    "CALLY",
    "ATION",
    "EING"
};
};
/*
*****/
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(int x, int y)
{
    srand(x);
    srandom(y);
}

/*
*****/
**
* create_random_a_string
*
** create a random alphanumeric string, of random length
between lo and
* hi and place them in designated buffer. Routine returns the
actual
* length.
* parameters
* -----

```

```

* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random alphanumeric string
*
*****
**
*/
int create_random_a_string( char *out_buffer, int length_lo, int
length_hi )
{
  int i, actual_length ;
  actual_length = rand_integer( length_lo, length_hi ) ;
  for (i = 0; i < actual_length; i++)
  {
    out_buffer[i] = alnum[rand_integer( 0, 61 )] ;
  }
  out_buffer[actual_length] = '\0' ;
  return (actual_length);
}

/*
*****
**
* create_random_n_string
*
* create a random numeric string, of random length between lo
and
* hi and place them in designated buffer. Routine returns the
actual
* length.
*
* parameters
* -----
*
* lo end of acceptable length range
* hi end of acceptable length range
*
* output
* -----
* actual length
* random numeric string
*
*****
**
*/
int create_random_n_string( char *out_buffer, int length_lo, int
length_hi )
{
  int i, actual_length ;
  actual_length = rand_integer( length_lo, length_hi ) ;
  for (i = 0; i < actual_length; i++)
  {
    out_buffer[i] = (char)rand_integer( 48,57 ) ;
  }
  out_buffer[actual_length] = '\0' ;
  return (actual_length);
}
/*

```

```

*****
**
* NUrand_val
*
* create a non-uniform random numeric value of type integer,
of random
* value between lo and hi. Number is NOT placed in BUFFER,
and IS
* simply RETURNED.
*
* Routine RETURNS the VALUE.
*
* parameters
* -----
* lo end of acceptable value range
* hi end of acceptable value range
*
* output
* -----
* random integer value RETURNED
*
*****
**
*/
int NUrand_val ( int A, int x, int y, int C )
{
  return((((rand_integer(0,A)|rand_integer(x,y))+C)%(y-x+1))+x);
}

/*
*****
**
* 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);
}
/*
*****
**
* create_a_string_with_original

```

```

*
* create a random alphanumeric string, of random length
between lo and
* hi and place them in designated buffer. Routine returns the
actual
* length.
*
* the word "ORIGINAL" is placed at a random location in the
buffer at
* random, for a given percent of the records.
*
* percent_to_set must be an integer value from 0 to 100.
* if 0, no records will be set. If 100, all records will be set.
*
* CANNOT USE ON STRINGS OF LENGTH LESS THAN 8 !
LOWER LIMIT MUST BE > 8 !
*
* parameters
* -----
* lo end of acceptable length range
* hi end of acceptable length range
* percentage of records to set to ORIGINAL
*
* output
* -----
* actual length
* random alphanumeric string with the word "ORIGINAL" is
placed at a
* random location
*
*****
**
*/
int create_a_string_with_original( char *out_buffer, int length_lo,
int length_hi, int percent_to_set )
{
  int actual_length, start_pos ;
  actual_length = create_random_a_string( out_buffer,
length_lo, length_hi ) ;
  if ( rand_integer( 1, 100 ) <= percent_to_set )
  {
    start_pos = rand_integer( 0, actual_length-8 ) ;
    strncpy(out_buffer+start_pos,"ORIGINAL",8) ;
  }
  return (actual_length);
}

/*
*****
**
* create_random_last_name
*
* parameters:
* out_buffer - target buffer for the generated last name
*
* description:
* create_random_last_name generates a random number
from 0 to 999
* inclusive. a random name is generated by associating a
random string
* with each digit of the generated number. the three strings
are

```

```

* concatenated to generate the name
*
*****
****/
int create_random_last_name(char *out_buffer, int cust_num)
{
    int random_num;
    if (cust_num == 0)
        random_num = 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/lval.h

#ifndef __LVAL_H

#define __LVAL_H
#define WAREHOUSES 256000
#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 - 2004
** 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
*/
#endif __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/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 - 2004
** 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(int x, int y);
int rand_integer( int val_lo, int val_hi );
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

```

Appendix - D: RTE Scripts

D.1 RTE Parameters

rteparams

```
/* C-Delta be the difference between C-LOAD and C-Run.
*/
/* C-Delta must be a value between 65..119 including the */
/* values of 65 and 119 and excluding the value of 96 and 112 */
#define MASTER_NUM1 0
#define MASTER_NUM2 0
#define MASTER_NUM3 0
#define MASTER_NUM4 0
#define MASTER_NUM5 1
#define MASTER_NUM6 0
#define MASTER_NUM7 0
#define MASTER_NUM8 0
#define MASTER_NUM9 0
#define MASTER_NUM10 0
#define MASTER_NUM11 0
#define MASTER_NUM12 0
#define MASTER_NUM13 0
#define MASTER_NUM14 0
#define MASTER_NUM15 0
#define MASTER_NUM16 0
#define MASTER_NUM17 0
#define MASTER_NUM18 0
#define MASTER_NUM19 0
#define MASTER_NUM20 0
#define MASTER_NUM21 0
#define MASTER_NUM22 0
#define MASTER_NUM23 0
#define MASTER_NUM24 0
#define MASTER_NUM25 0
#define MASTER_NUM26 0
#define MASTER_NUM27 0
#define MASTER_NUM28 0
#define MASTER_NUM29 0
#define MASTER_NUM30 0
#define MASTER_NUM31 0
#define MASTER_NUM32 0
#if MASTER_NUM1
MASTER "master1"
#elseif MASTER_NUM2
MASTER "master2"
#elseif MASTER_NUM3
MASTER "master3"
#elseif MASTER_NUM4
MASTER "master4"
#elseif MASTER_NUM5
MASTER "master5"
#elseif MASTER_NUM6
MASTER "master6"
#elseif MASTER_NUM7
MASTER "master7"
#elseif MASTER_NUM8
```

```
MASTER "master8"
#elseif MASTER_NUM9
MASTER "master9"
#elseif MASTER_NUM10
MASTER "master10"
#elseif MASTER_NUM11
MASTER "master11"
#elseif MASTER_NUM12
MASTER "master12"
#elseif MASTER_NUM13
MASTER "master13"
#elseif MASTER_NUM14
MASTER "master14"
#elseif MASTER_NUM15
MASTER "master15"
#elseif MASTER_NUM16
MASTER "master16"
#elseif MASTER_NUM17
MASTER "master17"
#elseif MASTER_NUM18
MASTER "master18"
#elseif MASTER_NUM19
MASTER "master19"
#elseif MASTER_NUM20
MASTER "master20"
#elseif MASTER_NUM21
MASTER "master21"
#elseif MASTER_NUM22
MASTER "master22"
#elseif MASTER_NUM23
MASTER "master23"
#elseif MASTER_NUM24
MASTER "master24"
#elseif MASTER_NUM25
MASTER "master25"
#elseif MASTER_NUM26
MASTER "master26"
#elseif MASTER_NUM27
MASTER "master27"
#elseif MASTER_NUM28
MASTER "master28"
#elseif MASTER_NUM29
MASTER "master29"
#elseif MASTER_NUM30
MASTER "master30"
#elseif MASTER_NUM31
MASTER "master31"
#elseif MASTER_NUM32
MASTER "master32"
#endif
/*---- SUT -----*/
SUT="toraus"
/*-----*/
LASTC=88
MEASUREMENT="1"
WAREHOUSES=256000
/*---- SLAVES -----*/
#if MASTER_NUM1
SLAVES driver1a, driver1b, driver1c, driver1d, driver2a,
driver2b, driver2c, driver2d, driver3a, driver3b, driver3c,
driver3d, driver4a, driver4b, driver4c, driver4d
#elseif MASTER_NUM2
```

```
SLAVES driver5a, driver5b, driver5c, driver5d, driver6a,
driver6b, driver6c, driver6d, driver7a, driver7b, driver7c,
driver7d, driver8a, driver8b, driver8c, driver8d
#elseif MASTER_NUM3
SLAVES driver9a, driver9b, driver9c, driver9d, driver10a,
driver10b, driver10c, driver10d, driver11a, driver11b, driver11c,
driver11d, driver12a, driver12b, driver12c, driver12d
#elseif MASTER_NUM4
SLAVES driver13a, driver13b, driver13c, driver13d, driver14a,
driver14b, driver14c, driver14d, driver15a, driver15b, driver15c,
driver15d, driver16a, driver16b, driver16c, driver16d
#elseif MASTER_NUM5
SLAVES driver17a, driver17b, driver17c, driver17d, driver18a,
driver18b, driver18c, driver18d, driver19a, driver19b, driver19c,
driver19d, driver20a, driver20b, driver20c, driver20d
#elseif MASTER_NUM6
SLAVES driver21a, driver21b, driver21c, driver21d, driver22a,
driver22b, driver22c, driver22d, driver23a, driver23b, driver23c,
driver23d, driver24a, driver24b, driver24c, driver24d
#elseif MASTER_NUM7
SLAVES driver25a, driver25b, driver25c, driver25d, driver26a,
driver26b, driver26c, driver26d, driver27a, driver27b, driver27c,
driver27d, driver28a, driver28b, driver28c, driver28d
#elseif MASTER_NUM8
SLAVES driver29a, driver29b, driver29c, driver29d, driver30a,
driver30b, driver30c, driver30d, driver31a, driver31b, driver31c,
driver31d, driver32a, driver32b, driver32c, driver32d
#elseif MASTER_NUM9
SLAVES driver33a, driver33b, driver33c, driver33d, driver34a,
driver34b, driver34c, driver34d, driver35a, driver35b, driver35c,
driver35d, driver36a, driver36b, driver36c, driver36d
#elseif MASTER_NUM10
SLAVES driver37a, driver37b, driver37c, driver37d, driver38a,
driver38b, driver38c, driver38d, driver39a, driver39b, driver39c,
driver39d, driver40a, driver40b, driver40c, driver40d
#elseif MASTER_NUM11
SLAVES driver41a, driver41b, driver41c, driver41d, driver42a,
driver42b, driver42c, driver42d, driver43a, driver43b, driver43c,
driver43d, driver44a, driver44b, driver44c, driver44d
#elseif MASTER_NUM12
SLAVES driver45a, driver45b, driver45c, driver45d, driver46a,
driver46b, driver46c, driver46d, driver47a, driver47b, driver47c,
driver47d, driver48a, driver48b, driver48c, driver48d
#elseif MASTER_NUM13
SLAVES driver49a, driver49b, driver49c, driver49d, driver50a,
driver50b, driver50c, driver50d, driver51a, driver51b, driver51c,
driver51d, driver52a, driver52b, driver52c, driver52d
#elseif MASTER_NUM14
SLAVES driver53a, driver53b, driver53c, driver53d, driver54a,
driver54b, driver54c, driver54d, driver55a, driver55b, driver55c,
driver55d, driver56a, driver56b, driver56c, driver56d
#elseif MASTER_NUM15
SLAVES driver57a, driver57b, driver57c, driver57d, driver58a,
driver58b, driver58c, driver58d, driver59a, driver59b, driver59c,
driver59d, driver60a, driver60b, driver60c, driver60d
#elseif MASTER_NUM16
SLAVES driver61a, driver61b, driver61c, driver61d, driver62a,
driver62b, driver62c, driver62d, driver63a, driver63b, driver63c,
driver63d, driver64a, driver64b, driver64c, driver64d
#elseif MASTER_NUM17
```

```

SLAVES driver65a, driver65b, driver65c, driver65d, driver66a,
driver66b, driver66c, driver66d, driver67a, driver67b, driver67c,
driver67d, driver68a, driver68b, driver68c, driver68d
#elif MASTER_NUM18
SLAVES driver69a, driver69b, driver69c, driver69d, driver70a,
driver70b, driver70c, driver70d, driver71a, driver71b, driver71c,
driver71d, driver72a, driver72b, driver72c, driver72d
#elif MASTER_NUM19
SLAVES driver73a, driver73b, driver73c, driver73d, driver74a,
driver74b, driver74c, driver74d, driver75a, driver75b, driver75c,
driver75d, driver76a, driver76b, driver76c, driver76d
#elif MASTER_NUM20
SLAVES driver77a, driver77b, driver77c, driver77d, driver78a,
driver78b, driver78c, driver78d, driver79a, driver79b, driver79c,
driver79d, driver80a, driver80b, driver80c, driver80d
#elif MASTER_NUM21
SLAVES driver81a, driver81b, driver81c, driver81d, driver82a,
driver82b, driver82c, driver82d, driver83a, driver83b, driver83c,
driver83d, driver84a, driver84b, driver84c, driver84d
#elif MASTER_NUM22
SLAVES driver85a, driver85b, driver85c, driver85d, driver86a,
driver86b, driver86c, driver86d, driver87a, driver87b, driver87c,
driver87d, driver88a, driver88b, driver88c, driver88d
#elif MASTER_NUM23
SLAVES driver89a, driver89b, driver89c, driver89d, driver90a,
driver90b, driver90c, driver90d, driver91a, driver91b, driver91c,
driver91d, driver92a, driver92b, driver92c, driver92d
#elif MASTER_NUM24
SLAVES driver93a, driver93b, driver93c, driver93d, driver94a,
driver94b, driver94c, driver94d, driver95a, driver95b, driver95c,
driver95d, driver96a, driver96b, driver96c, driver96d
#elif MASTER_NUM25
SLAVES driver97a, driver97b, driver97c, driver97d, driver98a,
driver98b, driver98c, driver98d, driver99a, driver99b, driver99c,
driver99d, driver100a, driver100b, driver100c, driver100d
#elif MASTER_NUM26
SLAVES driver101a, driver101b, driver101c, driver101d,
driver102a, driver102b, driver102c, driver102d, driver103a,
driver103b, driver103c, driver103d, driver104a, driver104b,
driver104c, driver104d
#elif MASTER_NUM27
SLAVES driver105a, driver105b, driver105c, driver105d,
driver106a, driver106b, driver106c, driver106d, driver107a,
driver107b, driver107c, driver107d, driver108a, driver108b,
driver108c, driver108d
#elif MASTER_NUM28
SLAVES driver109a, driver109b, driver109c, driver109d,
driver110a, driver110b, driver110c, driver110d, driver111a,
driver111b, driver111c, driver111d, driver112a, driver112b,
driver112c, driver112d
#elif MASTER_NUM29
SLAVES driver113a, driver113b, driver113c, driver113d,
driver114a, driver114b, driver114c, driver114d, driver115a,
driver115b, driver115c, driver115d, driver116a, driver116b,
driver116c, driver116d
#elif MASTER_NUM30
SLAVES driver117a, driver117b, driver117c, driver117d,
driver118a, driver118b, driver118c, driver118d, driver119a,
driver119b, driver119c, driver119d, driver120a, driver120b,
driver120c, driver120d
#elif MASTER_NUM31

```

```

SLAVES driver121a, driver121b, driver121c, driver121d,
driver122a, driver122b, driver122c, driver122d, driver123a,
driver123b, driver123c, driver123d, driver124a, driver124b,
driver124c, driver124d
#elif MASTER_NUM32
SLAVES driver125a, driver125b, driver125c, driver125d,
driver126a, driver126b, driver126c, driver126d, driver127a,
driver127b, driver127c, driver127d, driver128a, driver128b,
driver128c, driver128d
#endif
/* ---- CLIENTS -----*/
#if MASTER_NUM1
MAIN_CLIENT = client1
CLIENT_REAL = "client1 client2 client3 client4"
#elif MASTER_NUM2
MAIN_CLIENT = client5
CLIENT_REAL = "client5 client6 client7 client8"
#elif MASTER_NUM3
MAIN_CLIENT = client9
CLIENT_REAL = "client9 client10 client11 client12"
#elif MASTER_NUM4
MAIN_CLIENT = client13
CLIENT_REAL = "client13 client14 client15 client16"
#elif MASTER_NUM5
MAIN_CLIENT = client17
CLIENT_REAL = "client17 client18 client19 client20"
#elif MASTER_NUM6
MAIN_CLIENT = client21
CLIENT_REAL = "client21 client22 client23 client24"
#elif MASTER_NUM7
MAIN_CLIENT = client25
CLIENT_REAL = "client25 client26 client27 client28"
#elif MASTER_NUM8
MAIN_CLIENT = client29
CLIENT_REAL = "client29 client30 client31 client32"
#elif MASTER_NUM9
MAIN_CLIENT = client33
CLIENT_REAL = "client33 client34 client35 client36"
#elif MASTER_NUM10
MAIN_CLIENT = client37
CLIENT_REAL = "client37 client38 client39 client40"
#elif MASTER_NUM11
MAIN_CLIENT = client41
CLIENT_REAL = "client41 client42 client43 client44"
#elif MASTER_NUM12
MAIN_CLIENT = client45
CLIENT_REAL = "client45 client46 client47 client48"
#elif MASTER_NUM13
MAIN_CLIENT = client49
CLIENT_REAL = "client49 client50 client51 client52"
#elif MASTER_NUM14
MAIN_CLIENT = client53
CLIENT_REAL = "client53 client54 client55 client56"
#elif MASTER_NUM15
MAIN_CLIENT = client57
CLIENT_REAL = "client57 client58 client59 client60"
#elif MASTER_NUM16
MAIN_CLIENT = client61
CLIENT_REAL = "client61 client62 client63 client64"
#elif MASTER_NUM17
MAIN_CLIENT = client65
CLIENT_REAL = "client65 client66 client67 client68"

```

```

#elif MASTER_NUM18
MAIN_CLIENT = client69
CLIENT_REAL = "client69 client70 client71 client72"
#elif MASTER_NUM19
MAIN_CLIENT = client73
CLIENT_REAL = "client73 client74 client75 client76"
#elif MASTER_NUM20
MAIN_CLIENT = client77
CLIENT_REAL = "client77 client78 client79 client80"
#elif MASTER_NUM21
MAIN_CLIENT = client81
CLIENT_REAL = "client81 client82 client83 client84"
#elif MASTER_NUM22
MAIN_CLIENT = client85
CLIENT_REAL = "client85 client86 client87 client88"
#elif MASTER_NUM23
MAIN_CLIENT = client89
CLIENT_REAL = "client89 client90 client91 client92"
#elif MASTER_NUM24
MAIN_CLIENT = client93
CLIENT_REAL = "client93 client94 client95 client96"
#elif MASTER_NUM25
MAIN_CLIENT = client97
CLIENT_REAL = "client97 client98 client99 client100"
#elif MASTER_NUM26
MAIN_CLIENT = client101
CLIENT_REAL = "client101 client102 client103 client104"
#elif MASTER_NUM27
MAIN_CLIENT = client105
CLIENT_REAL = "client105 client106 client107 client108"
#elif MASTER_NUM28
MAIN_CLIENT = client109
CLIENT_REAL = "client109 client110 client111 client112"
#elif MASTER_NUM29
MAIN_CLIENT = client113
CLIENT_REAL = "client113 client114 client115 client116"
#elif MASTER_NUM30
MAIN_CLIENT = client117
CLIENT_REAL = "client117 client118 client119 client120"
#elif MASTER_NUM31
MAIN_CLIENT = client121
CLIENT_REAL = "client121 client122 client123 client124"
#elif MASTER_NUM32
MAIN_CLIENT = client125
CLIENT_REAL = "client125 client126 client127 client128"
#endif
/*---- more client ctuff -----*/
#if MASTER_NUM1
CLIENT client1a tpcc tpcc
CLIENT client2a tpcc tpcc
CLIENT client3a tpcc tpcc
CLIENT client4a tpcc tpcc
#elif MASTER_NUM2
CLIENT client5a tpcc tpcc
CLIENT client6a tpcc tpcc
CLIENT client7a tpcc tpcc
CLIENT client8a tpcc tpcc
#elif MASTER_NUM3
CLIENT client9a tpcc tpcc
CLIENT client10a tpcc tpcc
CLIENT client11a tpcc tpcc
CLIENT client12a tpcc tpcc

```

```

#elif MASTER_NUM4
CLIENT client13a tpcc tpcc
CLIENT client14a tpcc tpcc
CLIENT client15a tpcc tpcc
CLIENT client16a tpcc tpcc
#elif MASTER_NUM5
CLIENT client17a tpcc tpcc
CLIENT client18a tpcc tpcc
CLIENT client19a tpcc tpcc
CLIENT client20a tpcc tpcc
#elif MASTER_NUM6
CLIENT client21a tpcc tpcc
CLIENT client22a tpcc tpcc
CLIENT client23a tpcc tpcc
CLIENT client24a tpcc tpcc
#elif MASTER_NUM7
CLIENT client25a tpcc tpcc
CLIENT client26a tpcc tpcc
CLIENT client27a tpcc tpcc
CLIENT client28a tpcc tpcc
#elif MASTER_NUM8
CLIENT client29a tpcc tpcc
CLIENT client30a tpcc tpcc
CLIENT client31a tpcc tpcc
CLIENT client32a tpcc tpcc
#elif MASTER_NUM9
CLIENT client33a tpcc tpcc
CLIENT client34a tpcc tpcc
CLIENT client35a tpcc tpcc
CLIENT client36a tpcc tpcc
#elif MASTER_NUM10
CLIENT client37a tpcc tpcc
CLIENT client38a tpcc tpcc
CLIENT client39a tpcc tpcc
CLIENT client40a tpcc tpcc
#elif MASTER_NUM11
CLIENT client41a tpcc tpcc
CLIENT client42a tpcc tpcc
CLIENT client43a tpcc tpcc
CLIENT client44a tpcc tpcc
#elif MASTER_NUM12
CLIENT client45a tpcc tpcc
CLIENT client46a tpcc tpcc
CLIENT client47a tpcc tpcc
CLIENT client48a tpcc tpcc
#elif MASTER_NUM13
CLIENT client49a tpcc tpcc
CLIENT client50a tpcc tpcc
CLIENT client51a tpcc tpcc
CLIENT client52a tpcc tpcc
#elif MASTER_NUM14
CLIENT client53a tpcc tpcc
CLIENT client54a tpcc tpcc
CLIENT client55a tpcc tpcc
CLIENT client56a tpcc tpcc
#elif MASTER_NUM15
CLIENT client57a tpcc tpcc
CLIENT client58a tpcc tpcc
CLIENT client59a tpcc tpcc
CLIENT client60a tpcc tpcc
#elif MASTER_NUM16
CLIENT client61a tpcc tpcc

```

```

CLIENT client62a tpcc tpcc
CLIENT client63a tpcc tpcc
CLIENT client64a tpcc tpcc
#elif MASTER_NUM17
CLIENT client65a tpcc tpcc
CLIENT client66a tpcc tpcc
CLIENT client67a tpcc tpcc
CLIENT client68a tpcc tpcc
#elif MASTER_NUM18
CLIENT client69a tpcc tpcc
CLIENT client70a tpcc tpcc
CLIENT client71a tpcc tpcc
CLIENT client72a tpcc tpcc
#elif MASTER_NUM19
CLIENT client73a tpcc tpcc
CLIENT client74a tpcc tpcc
CLIENT client75a tpcc tpcc
CLIENT client76a tpcc tpcc
#elif MASTER_NUM20
CLIENT client77a tpcc tpcc
CLIENT client78a tpcc tpcc
CLIENT client79a tpcc tpcc
CLIENT client80a tpcc tpcc
#elif MASTER_NUM21
CLIENT client81a tpcc tpcc
CLIENT client82a tpcc tpcc
CLIENT client83a tpcc tpcc
CLIENT client84a tpcc tpcc
#elif MASTER_NUM22
CLIENT client85a tpcc tpcc
CLIENT client86a tpcc tpcc
CLIENT client87a tpcc tpcc
CLIENT client88a tpcc tpcc
#elif MASTER_NUM23
CLIENT client89a tpcc tpcc
CLIENT client90a tpcc tpcc
CLIENT client91a tpcc tpcc
CLIENT client92a tpcc tpcc
#elif MASTER_NUM24
CLIENT client93a tpcc tpcc
CLIENT client94a tpcc tpcc
CLIENT client95a tpcc tpcc
CLIENT client96a tpcc tpcc
#elif MASTER_NUM25
CLIENT client97a tpcc tpcc
CLIENT client98a tpcc tpcc
CLIENT client99a tpcc tpcc
CLIENT client100a tpcc tpcc
#elif MASTER_NUM26
CLIENT client101a tpcc tpcc
CLIENT client102a tpcc tpcc
CLIENT client103a tpcc tpcc
CLIENT client104a tpcc tpcc
#elif MASTER_NUM27
CLIENT client105a tpcc tpcc
CLIENT client106a tpcc tpcc
CLIENT client107a tpcc tpcc
CLIENT client108a tpcc tpcc
#elif MASTER_NUM28
CLIENT client109a tpcc tpcc
CLIENT client110a tpcc tpcc
CLIENT client111a tpcc tpcc

```

```

CLIENT client112a tpcc tpcc
#elif MASTER_NUM29
CLIENT client113a tpcc tpcc
CLIENT client114a tpcc tpcc
CLIENT client115a tpcc tpcc
CLIENT client116a tpcc tpcc
#elif MASTER_NUM30
CLIENT client117a tpcc tpcc
CLIENT client118a tpcc tpcc
CLIENT client119a tpcc tpcc
CLIENT client120a tpcc tpcc
#elif MASTER_NUM31
CLIENT client121a tpcc tpcc
CLIENT client122a tpcc tpcc
CLIENT client123a tpcc tpcc
CLIENT client124a tpcc tpcc
#elif MASTER_NUM32
CLIENT client125a tpcc tpcc
CLIENT client126a tpcc tpcc
CLIENT client127a tpcc tpcc
CLIENT client128a tpcc tpcc
#endif
/*-----*/
TELNET telnet 23
SOCKET socket 199703
/* --- Sockets -----*/
#if MASTER_NUM1
SOCKET_NETWORK socket1 80 driver1a
SOCKET_NETWORK socket2 80 driver1b
SOCKET_NETWORK socket3 80 driver1c
SOCKET_NETWORK socket4 80 driver1d
SOCKET_NETWORK socket5 80 driver1a
SOCKET_NETWORK socket6 80 driver1b
SOCKET_NETWORK socket7 80 driver1c
SOCKET_NETWORK socket8 80 driver1d
SOCKET_NETWORK socket9 80 driver1a
SOCKET_NETWORK socket10 80 driver1b
SOCKET_NETWORK socket11 80 driver1c
SOCKET_NETWORK socket12 80 driver1d
SOCKET_NETWORK socket13 80 driver1a
SOCKET_NETWORK socket14 80 driver1b
SOCKET_NETWORK socket15 80 driver1c
SOCKET_NETWORK socket16 80 driver1d
SOCKET_NETWORK socket17 80 driver1a
SOCKET_NETWORK socket18 80 driver1b
SOCKET_NETWORK socket19 80 driver1c
SOCKET_NETWORK socket20 80 driver1d
SOCKET_NETWORK socket21 80 driver1a
SOCKET_NETWORK socket22 80 driver1b
SOCKET_NETWORK socket23 80 driver1c
SOCKET_NETWORK socket24 80 driver1d
SOCKET_NETWORK socket25 80 driver1a
SOCKET_NETWORK socket26 80 driver1b
SOCKET_NETWORK socket27 80 driver1c
SOCKET_NETWORK socket28 80 driver1d
SOCKET_NETWORK socket29 80 driver1a
SOCKET_NETWORK socket30 80 driver1b
SOCKET_NETWORK socket31 80 driver1c
SOCKET_NETWORK socket32 80 driver1d
SOCKET_NETWORK socket33 80 driver2a
SOCKET_NETWORK socket34 80 driver2b
SOCKET_NETWORK socket35 80 driver2c

```



```

#endif
/*-----*/
OUTPUTNAME="regattaH"
CPU=48
#if 0
BEGIN_WAIT=5:00
RAMPUP=20:30
RUNTIME=30:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=20:00
#else
BEGIN_WAIT=25:00
RAMPUP=45:00
RUNTIME=150:00
RAMPDOWN_WAIT=2:00
RAMPDOWN=20:00
#endif
/*RAMPUP_SEC 2700 */
/*RUNTIME_SEC 9000 */
/*WAREHOUSES 256000 */
/*CLIENTS 3712 */
INTERVAL=1:00 /* Interval to calculate mix from */
LOGIN_MAX_LOAD = 3
LOGIN_BEGIN = 0 /* skip login state if set to 1 */
NOBEGIN = 1
KEYSTROKE_PACKET_SIZE = 0
MAX_CONCURRENT_SPAWN = 3
SPAWN_COUNT = 3
MIN_PORT = 8088
MAX_PORT = 8089
/* User variables. Think, Emulex Delay, %desired, %min, %max */
#if 1 /* Testing */
NEWORDER = "12.02, 0, 0"
PAYMENT = "12.02, 0, 0, 43.03, 43.03, 43.03 "
ORDSTAT = "10.01, 0, 0, 4.02, 4.02, 4.02 "
DELIVERY = "05.02, 0, 0, 4.02, 4.02, 4.02 "
STOCKLEV = "05.02, 0, 0, 4.02, 4.02, 4.02 "
#elseif 0 /* From rtparams.null */
NEWORDER = "12.25, 0.42, 0.38"
PAYMENT = "12.25, 0.19, 0.23, 43.2, 41.1, 45.3 "
ORDSTAT = "10.50, 0.39, 0.21, 4.1, 3.9, 4.3 "
DELIVERY = "05.5, 0.19, 0.15, 4.1, 3.9, 4.3 "
STOCKLEV = "05.5, 0.25, 0.18, 4.1, 3.9, 4.3 "
#elseif 0 /* From Pookeepsie */
NEWORDER = "16.25, 0.42, 0.38"
PAYMENT = "16.25, 0.19, 0.23, 43.15, 43.15, 43.15 "
ORDSTAT = "14.50, 0.39, 0.21, 4.03, 4.03, 4.03 "
DELIVERY = "09.50, 0.19, 0.15, 4.03, 4.03, 4.03 "
STOCKLEV = "09.50, 0.25, 0.18, 4.03, 4.03, 4.03 "
#endif
/*---- Starting users on sockets -----*/
#if MASTER_NUM1
START_RANGE client1a socket1 620 0-62
START_RANGE client1a socket2 630 62-125
START_RANGE client1a socket3 620 125-187
START_RANGE client1a socket4 630 187-250
START_RANGE client1a socket5 620 250-312
START_RANGE client1a socket6 630 312-375
START_RANGE client1a socket7 620 375-437
START_RANGE client1a socket8 630 437-500
START_RANGE client1a socket9 620 500-562

```

```

START_RANGE client1a socket10 630 562-625
START_RANGE client1a socket11 620 625-687
START_RANGE client1a socket12 630 687-750
START_RANGE client1a socket13 620 750-812
START_RANGE client1a socket14 630 812-875
START_RANGE client1a socket15 620 875-937
START_RANGE client1a socket16 630 937-1000
START_RANGE client1a socket17 620 1000-1062
START_RANGE client1a socket18 630 1062-1125
START_RANGE client1a socket19 620 1125-1187
START_RANGE client1a socket20 630 1187-1250
START_RANGE client1a socket21 620 1250-1312
START_RANGE client1a socket22 630 1312-1375
START_RANGE client1a socket23 620 1375-1437
START_RANGE client1a socket24 630 1437-1500
START_RANGE client1a socket25 620 1500-1562
START_RANGE client1a socket26 630 1562-1625
START_RANGE client1a socket27 620 1625-1687
START_RANGE client1a socket28 630 1687-1750
START_RANGE client1a socket29 620 1750-1812
START_RANGE client1a socket30 630 1812-1875
START_RANGE client1a socket31 620 1875-1937
START_RANGE client1a socket32 630 1937-2000
START_RANGE client2a socket33 620 2000-2062
START_RANGE client2a socket34 630 2062-2125
START_RANGE client2a socket35 620 2125-2187
START_RANGE client2a socket36 630 2187-2250
START_RANGE client2a socket37 620 2250-2312
START_RANGE client2a socket38 630 2312-2375
START_RANGE client2a socket39 620 2375-2437
START_RANGE client2a socket40 630 2437-2500
START_RANGE client2a socket41 620 2500-2562
START_RANGE client2a socket42 630 2562-2625
START_RANGE client2a socket43 620 2625-2687
START_RANGE client2a socket44 630 2687-2750
START_RANGE client2a socket45 620 2750-2812
START_RANGE client2a socket46 630 2812-2875
START_RANGE client2a socket47 620 2875-2937
START_RANGE client2a socket48 630 2937-3000
START_RANGE client2a socket49 620 3000-3062
START_RANGE client2a socket50 630 3062-3125
START_RANGE client2a socket51 620 3125-3187
START_RANGE client2a socket52 630 3187-3250
START_RANGE client2a socket53 620 3250-3312
START_RANGE client2a socket54 630 3312-3375
START_RANGE client2a socket55 620 3375-3437
START_RANGE client2a socket56 630 3437-3500
START_RANGE client2a socket57 620 3500-3562
START_RANGE client2a socket58 630 3562-3625
START_RANGE client2a socket59 620 3625-3687
START_RANGE client2a socket60 630 3687-3750
START_RANGE client2a socket61 620 3750-3812
START_RANGE client2a socket62 630 3812-3875
START_RANGE client2a socket63 620 3875-3937
START_RANGE client2a socket64 630 3937-4000
START_RANGE client3a socket65 620 4000-4062
START_RANGE client3a socket66 630 4062-4125
START_RANGE client3a socket67 620 4125-4187
START_RANGE client3a socket68 630 4187-4250
START_RANGE client3a socket69 620 4250-4312
START_RANGE client3a socket70 630 4312-4375
START_RANGE client3a socket71 620 4375-4437

```

```

START_RANGE client3a socket72 630 4437-4500
START_RANGE client3a socket73 620 4500-4562
START_RANGE client3a socket74 630 4562-4625
START_RANGE client3a socket75 620 4625-4687
START_RANGE client3a socket76 630 4687-4750
START_RANGE client3a socket77 620 4750-4812
START_RANGE client3a socket78 630 4812-4875
START_RANGE client3a socket79 620 4875-4937
START_RANGE client3a socket80 630 4937-5000
START_RANGE client3a socket81 620 5000-5062
START_RANGE client3a socket82 630 5062-5125
START_RANGE client3a socket83 620 5125-5187
START_RANGE client3a socket84 630 5187-5250
START_RANGE client3a socket85 620 5250-5312
START_RANGE client3a socket86 630 5312-5375
START_RANGE client3a socket87 620 5375-5437
START_RANGE client3a socket88 630 5437-5500
START_RANGE client3a socket89 620 5500-5562
START_RANGE client3a socket90 630 5562-5625
START_RANGE client3a socket91 620 5625-5687
START_RANGE client3a socket92 630 5687-5750
START_RANGE client3a socket93 620 5750-5812
START_RANGE client3a socket94 630 5812-5875
START_RANGE client3a socket95 620 5875-5937
START_RANGE client3a socket96 630 5937-6000
START_RANGE client4a socket97 620 6000-6062
START_RANGE client4a socket98 630 6062-6125
START_RANGE client4a socket99 620 6125-6187
START_RANGE client4a socket100 630 6187-6250
START_RANGE client4a socket101 620 6250-6312
START_RANGE client4a socket102 630 6312-6375
START_RANGE client4a socket103 620 6375-6437
START_RANGE client4a socket104 630 6437-6500
START_RANGE client4a socket105 620 6500-6562
START_RANGE client4a socket106 630 6562-6625
START_RANGE client4a socket107 620 6625-6687
START_RANGE client4a socket108 630 6687-6750
START_RANGE client4a socket109 620 6750-6812
START_RANGE client4a socket110 630 6812-6875
START_RANGE client4a socket111 620 6875-6937
START_RANGE client4a socket112 620 6937-7000
START_RANGE client4a socket113 620 7000-7062
START_RANGE client4a socket114 630 7062-7125
START_RANGE client4a socket115 620 7125-7187
START_RANGE client4a socket116 630 7187-7250
START_RANGE client4a socket117 620 7250-7312
START_RANGE client4a socket118 630 7312-7375
START_RANGE client4a socket119 620 7375-7437
START_RANGE client4a socket120 630 7437-7500
START_RANGE client4a socket121 620 7500-7562
START_RANGE client4a socket122 630 7562-7625
START_RANGE client4a socket123 620 7625-7687
START_RANGE client4a socket124 630 7687-7750
START_RANGE client4a socket125 620 7750-7812
START_RANGE client4a socket126 630 7812-7875
START_RANGE client4a socket127 620 7875-7937
START_RANGE client4a socket128 630 7937-8000
#elseif MASTER_NUM2
START_RANGE client5a socket129 620 8000-8062
START_RANGE client5a socket130 630 8062-8125
START_RANGE client5a socket131 620 8125-8187
START_RANGE client5a socket132 630 8187-8250

```

START_RANGE client5a socket133 620 8250-8312
START_RANGE client5a socket134 630 8312-8375
START_RANGE client5a socket135 620 8375-8437
START_RANGE client5a socket136 630 8437-8500
START_RANGE client5a socket137 620 8500-8562
START_RANGE client5a socket138 630 8562-8625
START_RANGE client5a socket139 620 8625-8687
START_RANGE client5a socket140 630 8687-8750
START_RANGE client5a socket141 620 8750-8812
START_RANGE client5a socket142 630 8812-8875
START_RANGE client5a socket143 620 8875-8937
START_RANGE client5a socket144 630 8937-9000
START_RANGE client5a socket145 620 9000-9062
START_RANGE client5a socket146 630 9062-9125
START_RANGE client5a socket147 620 9125-9187
START_RANGE client5a socket148 630 9187-9250
START_RANGE client5a socket149 620 9250-9312
START_RANGE client5a socket150 630 9312-9375
START_RANGE client5a socket151 620 9375-9437
START_RANGE client5a socket152 630 9437-9500
START_RANGE client5a socket153 620 9500-9562
START_RANGE client5a socket154 630 9562-9625
START_RANGE client5a socket155 620 9625-9687
START_RANGE client5a socket156 630 9687-9750
START_RANGE client5a socket157 620 9750-9812
START_RANGE client5a socket158 630 9812-9875
START_RANGE client5a socket159 620 9875-9937
START_RANGE client5a socket160 630 9937-10000
START_RANGE client6a socket161 620 10000-10062
START_RANGE client6a socket162 630 10062-10125
START_RANGE client6a socket163 620 10125-10187
START_RANGE client6a socket164 630 10187-10250
START_RANGE client6a socket165 620 10250-10312
START_RANGE client6a socket166 630 10312-10375
START_RANGE client6a socket167 620 10375-10437
START_RANGE client6a socket168 630 10437-10500
START_RANGE client6a socket169 620 10500-10562
START_RANGE client6a socket170 630 10562-10625
START_RANGE client6a socket171 620 10625-10687
START_RANGE client6a socket172 630 10687-10750
START_RANGE client6a socket173 620 10750-10812
START_RANGE client6a socket174 630 10812-10875
START_RANGE client6a socket175 620 10875-10937
START_RANGE client6a socket176 630 10937-11000
START_RANGE client6a socket177 620 11000-11062
START_RANGE client6a socket178 630 11062-11125
START_RANGE client6a socket179 620 11125-11187
START_RANGE client6a socket180 630 11187-11250
START_RANGE client6a socket181 620 11250-11312
START_RANGE client6a socket182 630 11312-11375
START_RANGE client6a socket183 620 11375-11437
START_RANGE client6a socket184 630 11437-11500
START_RANGE client6a socket185 620 11500-11562
START_RANGE client6a socket186 630 11562-11625
START_RANGE client6a socket187 620 11625-11687
START_RANGE client6a socket188 630 11687-11750
START_RANGE client6a socket189 620 11750-11812
START_RANGE client6a socket190 630 11812-11875
START_RANGE client6a socket191 620 11875-11937
START_RANGE client6a socket192 630 11937-12000
START_RANGE client7a socket193 620 12000-12062
START_RANGE client7a socket194 630 12062-12125

START_RANGE client7a socket195 620 12125-12187
START_RANGE client7a socket196 630 12187-12250
START_RANGE client7a socket197 620 12250-12312
START_RANGE client7a socket198 630 12312-12375
START_RANGE client7a socket199 620 12375-12437
START_RANGE client7a socket200 630 12437-12500
START_RANGE client7a socket201 620 12500-12562
START_RANGE client7a socket202 630 12562-12625
START_RANGE client7a socket203 620 12625-12687
START_RANGE client7a socket204 630 12687-12750
START_RANGE client7a socket205 620 12750-12812
START_RANGE client7a socket206 630 12812-12875
START_RANGE client7a socket207 620 12875-12937
START_RANGE client7a socket208 630 12937-13000
START_RANGE client7a socket209 620 13000-13062
START_RANGE client7a socket210 630 13062-13125
START_RANGE client7a socket211 620 13125-13187
START_RANGE client7a socket212 630 13187-13250
START_RANGE client7a socket213 620 13250-13312
START_RANGE client7a socket214 630 13312-13375
START_RANGE client7a socket215 620 13375-13437
START_RANGE client7a socket216 630 13437-13500
START_RANGE client7a socket217 620 13500-13562
START_RANGE client7a socket218 630 13562-13625
START_RANGE client7a socket219 620 13625-13687
START_RANGE client7a socket220 630 13687-13750
START_RANGE client7a socket221 620 13750-13812
START_RANGE client7a socket222 630 13812-13875
START_RANGE client7a socket223 620 13875-13937
START_RANGE client7a socket224 630 13937-14000
START_RANGE client8a socket225 620 14000-14062
START_RANGE client8a socket226 630 14062-14125
START_RANGE client8a socket227 620 14125-14187
START_RANGE client8a socket228 630 14187-14250
START_RANGE client8a socket229 620 14250-14312
START_RANGE client8a socket230 630 14312-14375
START_RANGE client8a socket231 620 14375-14437
START_RANGE client8a socket232 630 14437-14500
START_RANGE client8a socket233 620 14500-14562
START_RANGE client8a socket234 630 14562-14625
START_RANGE client8a socket235 620 14625-14687
START_RANGE client8a socket236 630 14687-14750
START_RANGE client8a socket237 620 14750-14812
START_RANGE client8a socket238 630 14812-14875
START_RANGE client8a socket239 620 14875-14937
START_RANGE client8a socket240 630 14937-15000
START_RANGE client8a socket241 620 15000-15062
START_RANGE client8a socket242 630 15062-15125
START_RANGE client8a socket243 620 15125-15187
START_RANGE client8a socket244 630 15187-15250
START_RANGE client8a socket245 620 15250-15312
START_RANGE client8a socket246 630 15312-15375
START_RANGE client8a socket247 620 15375-15437
START_RANGE client8a socket248 630 15437-15500
START_RANGE client8a socket249 620 15500-15562
START_RANGE client8a socket250 630 15562-15625
START_RANGE client8a socket251 620 15625-15687
START_RANGE client8a socket252 630 15687-15750
START_RANGE client8a socket253 620 15750-15812
START_RANGE client8a socket254 630 15812-15875
START_RANGE client8a socket255 620 15875-15937
START_RANGE client8a socket256 630 15937-16000

#elif MASTER_NUM3
START_RANGE client9a socket257 620 16000-16062
START_RANGE client9a socket258 630 16062-16125
START_RANGE client9a socket259 620 16125-16187
START_RANGE client9a socket260 630 16187-16250
START_RANGE client9a socket261 620 16250-16312
START_RANGE client9a socket262 630 16312-16375
START_RANGE client9a socket263 620 16375-16437
START_RANGE client9a socket264 630 16437-16500
START_RANGE client9a socket265 620 16500-16562
START_RANGE client9a socket266 630 16562-16625
START_RANGE client9a socket267 620 16625-16687
START_RANGE client9a socket268 630 16687-16750
START_RANGE client9a socket269 620 16750-16812
START_RANGE client9a socket270 630 16812-16875
START_RANGE client9a socket271 620 16875-16937
START_RANGE client9a socket272 630 16937-17000
START_RANGE client9a socket273 620 17000-17062
START_RANGE client9a socket274 630 17062-17125
START_RANGE client9a socket275 620 17125-17187
START_RANGE client9a socket276 630 17187-17250
START_RANGE client9a socket277 620 17250-17312
START_RANGE client9a socket278 630 17312-17375
START_RANGE client9a socket279 620 17375-17437
START_RANGE client9a socket280 630 17437-17500
START_RANGE client9a socket281 620 17500-17562
START_RANGE client9a socket282 630 17562-17625
START_RANGE client9a socket283 620 17625-17687
START_RANGE client9a socket284 630 17687-17750
START_RANGE client9a socket285 620 17750-17812
START_RANGE client9a socket286 630 17812-17875
START_RANGE client9a socket287 620 17875-17937
START_RANGE client9a socket288 630 17937-18000
START_RANGE client10a socket289 620 18000-18062
START_RANGE client10a socket290 630 18062-18125
START_RANGE client10a socket291 620 18125-18187
START_RANGE client10a socket292 630 18187-18250
START_RANGE client10a socket293 620 18250-18312
START_RANGE client10a socket294 630 18312-18375
START_RANGE client10a socket295 620 18375-18437
START_RANGE client10a socket296 630 18437-18500
START_RANGE client10a socket297 620 18500-18562
START_RANGE client10a socket298 630 18562-18625
START_RANGE client10a socket299 620 18625-18687
START_RANGE client10a socket300 630 18687-18750
START_RANGE client10a socket301 620 18750-18812
START_RANGE client10a socket302 630 18812-18875
START_RANGE client10a socket303 620 18875-18937
START_RANGE client10a socket304 630 18937-19000
START_RANGE client10a socket305 620 19000-19062
START_RANGE client10a socket306 630 19062-19125
START_RANGE client10a socket307 620 19125-19187
START_RANGE client10a socket308 630 19187-19250
START_RANGE client10a socket309 620 19250-19312
START_RANGE client10a socket310 630 19312-19375
START_RANGE client10a socket311 620 19375-19437
START_RANGE client10a socket312 630 19437-19500
START_RANGE client10a socket313 620 19500-19562
START_RANGE client10a socket314 630 19562-19625
START_RANGE client10a socket315 620 19625-19687
START_RANGE client10a socket316 630 19687-19750
START_RANGE client10a socket317 620 19750-19812

START_RANGE client10a socket318 630 19812-19875
START_RANGE client10a socket319 620 19875-19937
START_RANGE client10a socket320 630 19937-20000
START_RANGE client11a socket321 620 20000-20062
START_RANGE client11a socket322 630 20062-20125
START_RANGE client11a socket323 620 20125-20187
START_RANGE client11a socket324 630 20187-20250
START_RANGE client11a socket325 620 20250-20312
START_RANGE client11a socket326 630 20312-20375
START_RANGE client11a socket327 620 20375-20437
START_RANGE client11a socket328 630 20437-20500
START_RANGE client11a socket329 620 20500-20562
START_RANGE client11a socket330 630 20562-20625
START_RANGE client11a socket331 620 20625-20687
START_RANGE client11a socket332 630 20687-20750
START_RANGE client11a socket333 620 20750-20812
START_RANGE client11a socket334 630 20812-20875
START_RANGE client11a socket335 620 20875-20937
START_RANGE client11a socket336 630 20937-21000
START_RANGE client11a socket337 620 21000-21062
START_RANGE client11a socket338 630 21062-21125
START_RANGE client11a socket339 620 21125-21187
START_RANGE client11a socket340 630 21187-21250
START_RANGE client11a socket341 620 21250-21312
START_RANGE client11a socket342 630 21312-21375
START_RANGE client11a socket343 620 21375-21437
START_RANGE client11a socket344 630 21437-21500
START_RANGE client11a socket345 620 21500-21562
START_RANGE client11a socket346 630 21562-21625
START_RANGE client11a socket347 620 21625-21687
START_RANGE client11a socket348 630 21687-21750
START_RANGE client11a socket349 620 21750-21812
START_RANGE client11a socket350 630 21812-21875
START_RANGE client11a socket351 620 21875-21937
START_RANGE client11a socket352 630 21937-22000
START_RANGE client12a socket353 620 22000-22062
START_RANGE client12a socket354 630 22062-22125
START_RANGE client12a socket355 620 22125-22187
START_RANGE client12a socket356 630 22187-22250
START_RANGE client12a socket357 620 22250-22312
START_RANGE client12a socket358 630 22312-22375
START_RANGE client12a socket359 620 22375-22437
START_RANGE client12a socket360 630 22437-22500
START_RANGE client12a socket361 620 22500-22562
START_RANGE client12a socket362 630 22562-22625
START_RANGE client12a socket363 620 22625-22687
START_RANGE client12a socket364 630 22687-22750
START_RANGE client12a socket365 620 22750-22812
START_RANGE client12a socket366 630 22812-22875
START_RANGE client12a socket367 620 22875-22937
START_RANGE client12a socket368 630 22937-23000
START_RANGE client12a socket369 620 23000-23062
START_RANGE client12a socket370 630 23062-23125
START_RANGE client12a socket371 620 23125-23187
START_RANGE client12a socket372 630 23187-23250
START_RANGE client12a socket373 620 23250-23312
START_RANGE client12a socket374 630 23312-23375
START_RANGE client12a socket375 620 23375-23437
START_RANGE client12a socket376 630 23437-23500
START_RANGE client12a socket377 620 23500-23562
START_RANGE client12a socket378 630 23562-23625
START_RANGE client12a socket379 620 23625-23687

START_RANGE client12a socket380 630 23687-23750
START_RANGE client12a socket381 620 23750-23812
START_RANGE client12a socket382 630 23812-23875
START_RANGE client12a socket383 620 23875-23937
START_RANGE client12a socket384 630 23937-24000
#elif MASTER_NUM4
START_RANGE client13a socket385 620 24000-24062
START_RANGE client13a socket386 630 24062-24125
START_RANGE client13a socket387 620 24125-24187
START_RANGE client13a socket388 630 24187-24250
START_RANGE client13a socket389 620 24250-24312
START_RANGE client13a socket390 630 24312-24375
START_RANGE client13a socket391 620 24375-24437
START_RANGE client13a socket392 630 24437-24500
START_RANGE client13a socket393 620 24500-24562
START_RANGE client13a socket394 630 24562-24625
START_RANGE client13a socket395 620 24625-24687
START_RANGE client13a socket396 630 24687-24750
START_RANGE client13a socket397 620 24750-24812
START_RANGE client13a socket398 630 24812-24875
START_RANGE client13a socket399 620 24875-24937
START_RANGE client13a socket400 630 24937-25000
START_RANGE client13a socket401 620 25000-25062
START_RANGE client13a socket402 630 25062-25125
START_RANGE client13a socket403 620 25125-25187
START_RANGE client13a socket404 630 25187-25250
START_RANGE client13a socket405 620 25250-25312
START_RANGE client13a socket406 630 25312-25375
START_RANGE client13a socket407 620 25375-25437
START_RANGE client13a socket408 630 25437-25500
START_RANGE client13a socket409 620 25500-25562
START_RANGE client13a socket410 630 25562-25625
START_RANGE client13a socket411 620 25625-25687
START_RANGE client13a socket412 630 25687-25750
START_RANGE client13a socket413 620 25750-25812
START_RANGE client13a socket414 630 25812-25875
START_RANGE client13a socket415 620 25875-25937
START_RANGE client13a socket416 630 25937-26000
START_RANGE client14a socket417 620 26000-26062
START_RANGE client14a socket418 630 26062-26125
START_RANGE client14a socket419 620 26125-26187
START_RANGE client14a socket420 630 26187-26250
START_RANGE client14a socket421 620 26250-26312
START_RANGE client14a socket422 630 26312-26375
START_RANGE client14a socket423 620 26375-26437
START_RANGE client14a socket424 630 26437-26500
START_RANGE client14a socket425 620 26500-26562
START_RANGE client14a socket426 630 26562-26625
START_RANGE client14a socket427 620 26625-26687
START_RANGE client14a socket428 630 26687-26750
START_RANGE client14a socket429 620 26750-26812
START_RANGE client14a socket430 630 26812-26875
START_RANGE client14a socket431 620 26875-26937
START_RANGE client14a socket432 630 26937-27000
START_RANGE client14a socket433 620 27000-27062
START_RANGE client14a socket434 630 27062-27125
START_RANGE client14a socket435 620 27125-27187
START_RANGE client14a socket436 630 27187-27250
START_RANGE client14a socket437 620 27250-27312
START_RANGE client14a socket438 630 27312-27375
START_RANGE client14a socket439 620 27375-27437
START_RANGE client14a socket440 630 27437-27500

START_RANGE client14a socket441 620 27500-27562
START_RANGE client14a socket442 630 27562-27625
START_RANGE client14a socket443 620 27625-27687
START_RANGE client14a socket444 630 27687-27750
START_RANGE client14a socket445 620 27750-27812
START_RANGE client14a socket446 630 27812-27875
START_RANGE client14a socket447 620 27875-27937
START_RANGE client14a socket448 630 27937-28000
START_RANGE client15a socket449 620 28000-28062
START_RANGE client15a socket450 630 28062-28125
START_RANGE client15a socket451 620 28125-28187
START_RANGE client15a socket452 630 28187-28250
START_RANGE client15a socket453 620 28250-28312
START_RANGE client15a socket454 630 28312-28375
START_RANGE client15a socket455 620 28375-28437
START_RANGE client15a socket456 630 28437-28500
START_RANGE client15a socket457 620 28500-28562
START_RANGE client15a socket458 630 28562-28625
START_RANGE client15a socket459 620 28625-28687
START_RANGE client15a socket460 630 28687-28750
START_RANGE client15a socket461 620 28750-28812
START_RANGE client15a socket462 630 28812-28875
START_RANGE client15a socket463 620 28875-28937
START_RANGE client15a socket464 630 28937-29000
START_RANGE client15a socket465 620 29000-29062
START_RANGE client15a socket466 630 29062-29125
START_RANGE client15a socket467 620 29125-29187
START_RANGE client15a socket468 630 29187-29250
START_RANGE client15a socket469 620 29250-29312
START_RANGE client15a socket470 630 29312-29375
START_RANGE client15a socket471 620 29375-29437
START_RANGE client15a socket472 630 29437-29500
START_RANGE client15a socket473 620 29500-29562
START_RANGE client15a socket474 630 29562-29625
START_RANGE client15a socket475 620 29625-29687
START_RANGE client15a socket476 630 29687-29750
START_RANGE client15a socket477 620 29750-29812
START_RANGE client15a socket478 630 29812-29875
START_RANGE client15a socket479 620 29875-29937
START_RANGE client15a socket480 630 29937-30000
START_RANGE client16a socket481 620 30000-30062
START_RANGE client16a socket482 630 30062-30125
START_RANGE client16a socket483 620 30125-30187
START_RANGE client16a socket484 630 30187-30250
START_RANGE client16a socket485 620 30250-30312
START_RANGE client16a socket486 630 30312-30375
START_RANGE client16a socket487 620 30375-30437
START_RANGE client16a socket488 630 30437-30500
START_RANGE client16a socket489 620 30500-30562
START_RANGE client16a socket490 630 30562-30625
START_RANGE client16a socket491 620 30625-30687
START_RANGE client16a socket492 630 30687-30750
START_RANGE client16a socket493 620 30750-30812
START_RANGE client16a socket494 630 30812-30875
START_RANGE client16a socket495 620 30875-30937
START_RANGE client16a socket496 630 30937-31000
START_RANGE client16a socket497 620 31000-31062
START_RANGE client16a socket498 630 31062-31125
START_RANGE client16a socket499 620 31125-31187
START_RANGE client16a socket500 630 31187-31250
START_RANGE client16a socket501 620 31250-31312
START_RANGE client16a socket502 630 31312-31375

START_RANGE client16a socket503 620 31375-31437
START_RANGE client16a socket504 630 31437-31500
START_RANGE client16a socket505 620 31500-31562
START_RANGE client16a socket506 630 31562-31625
START_RANGE client16a socket507 620 31625-31687
START_RANGE client16a socket508 630 31687-31750
START_RANGE client16a socket509 620 31750-31812
START_RANGE client16a socket510 630 31812-31875
START_RANGE client16a socket511 620 31875-31937
START_RANGE client16a socket512 630 31937-32000
#elif MASTER_NUM5
START_RANGE client17a socket513 620 32000-32062
START_RANGE client17a socket514 630 32062-32125
START_RANGE client17a socket515 620 32125-32187
START_RANGE client17a socket516 630 32187-32250
START_RANGE client17a socket517 620 32250-32312
START_RANGE client17a socket518 630 32312-32375
START_RANGE client17a socket519 620 32375-32437
START_RANGE client17a socket520 630 32437-32500
START_RANGE client17a socket521 620 32500-32562
START_RANGE client17a socket522 630 32562-32625
START_RANGE client17a socket523 620 32625-32687
START_RANGE client17a socket524 630 32687-32750
START_RANGE client17a socket525 620 32750-32812
START_RANGE client17a socket526 630 32812-32875
START_RANGE client17a socket527 620 32875-32937
START_RANGE client17a socket528 630 32937-33000
START_RANGE client17a socket529 620 33000-33062
START_RANGE client17a socket530 630 33062-33125
START_RANGE client17a socket531 620 33125-33187
START_RANGE client17a socket532 630 33187-33250
START_RANGE client17a socket533 620 33250-33312
START_RANGE client17a socket534 630 33312-33375
START_RANGE client17a socket535 620 33375-33437
START_RANGE client17a socket536 630 33437-33500
START_RANGE client17a socket537 620 33500-33562
START_RANGE client17a socket538 630 33562-33625
START_RANGE client17a socket539 620 33625-33687
START_RANGE client17a socket540 630 33687-33750
START_RANGE client17a socket541 620 33750-33812
START_RANGE client17a socket542 630 33812-33875
START_RANGE client17a socket543 620 33875-33937
START_RANGE client17a socket544 630 33937-34000
START_RANGE client17a socket545 620 34000-34062
START_RANGE client18a socket546 630 34062-34125
START_RANGE client18a socket547 620 34125-34187
START_RANGE client18a socket548 630 34187-34250
START_RANGE client18a socket549 620 34250-34312
START_RANGE client18a socket550 630 34312-34375
START_RANGE client18a socket551 620 34375-34437
START_RANGE client18a socket552 630 34437-34500
START_RANGE client18a socket553 620 34500-34562
START_RANGE client18a socket554 630 34562-34625
START_RANGE client18a socket555 620 34625-34687
START_RANGE client18a socket556 630 34687-34750
START_RANGE client18a socket557 620 34750-34812
START_RANGE client18a socket558 630 34812-34875
START_RANGE client18a socket559 620 34875-34937
START_RANGE client18a socket560 630 34937-35000
START_RANGE client18a socket561 620 35000-35062
START_RANGE client18a socket562 630 35062-35125
START_RANGE client18a socket563 620 35125-35187

START_RANGE client18a socket564 630 35187-35250
START_RANGE client18a socket565 620 35250-35312
START_RANGE client18a socket566 630 35312-35375
START_RANGE client18a socket567 620 35375-35437
START_RANGE client18a socket568 630 35437-35500
START_RANGE client18a socket569 620 35500-35562
START_RANGE client18a socket570 630 35562-35625
START_RANGE client18a socket571 620 35625-35687
START_RANGE client18a socket572 630 35687-35750
START_RANGE client18a socket573 620 35750-35812
START_RANGE client18a socket574 630 35812-35875
START_RANGE client18a socket575 620 35875-35937
START_RANGE client18a socket576 630 35937-36000
START_RANGE client19a socket577 620 36000-36062
START_RANGE client19a socket578 630 36062-36125
START_RANGE client19a socket579 620 36125-36187
START_RANGE client19a socket580 630 36187-36250
START_RANGE client19a socket581 620 36250-36312
START_RANGE client19a socket582 630 36312-36375
START_RANGE client19a socket583 620 36375-36437
START_RANGE client19a socket584 630 36437-36500
START_RANGE client19a socket585 620 36500-36562
START_RANGE client19a socket586 630 36562-36625
START_RANGE client19a socket587 620 36625-36687
START_RANGE client19a socket588 630 36687-36750
START_RANGE client19a socket589 620 36750-36812
START_RANGE client19a socket590 630 36812-36875
START_RANGE client19a socket591 620 36875-36937
START_RANGE client19a socket592 630 36937-37000
START_RANGE client19a socket593 620 37000-37062
START_RANGE client19a socket594 630 37062-37125
START_RANGE client19a socket595 620 37125-37187
START_RANGE client19a socket596 630 37187-37250
START_RANGE client19a socket597 620 37250-37312
START_RANGE client19a socket598 630 37312-37375
START_RANGE client19a socket599 620 37375-37437
START_RANGE client19a socket600 630 37437-37500
START_RANGE client19a socket601 620 37500-37562
START_RANGE client19a socket602 630 37562-37625
START_RANGE client19a socket603 620 37625-37687
START_RANGE client19a socket604 630 37687-37750
START_RANGE client19a socket605 620 37750-37812
START_RANGE client19a socket606 630 37812-37875
START_RANGE client19a socket607 620 37875-37937
START_RANGE client19a socket608 630 37937-38000
START_RANGE client20a socket609 620 38000-38062
START_RANGE client20a socket610 630 38062-38125
START_RANGE client20a socket611 620 38125-38187
START_RANGE client20a socket612 630 38187-38250
START_RANGE client20a socket613 620 38250-38312
START_RANGE client20a socket614 630 38312-38375
START_RANGE client20a socket615 620 38375-38437
START_RANGE client20a socket616 630 38437-38500
START_RANGE client20a socket617 620 38500-38562
START_RANGE client20a socket618 630 38562-38625
START_RANGE client20a socket619 620 38625-38687
START_RANGE client20a socket620 630 38687-38750
START_RANGE client20a socket621 620 38750-38812
START_RANGE client20a socket622 630 38812-38875
START_RANGE client20a socket623 620 38875-38937
START_RANGE client20a socket624 630 38937-39000
START_RANGE client20a socket625 620 39000-39062

START_RANGE client20a socket626 630 39062-39125
START_RANGE client20a socket627 620 39125-39187
START_RANGE client20a socket628 630 39187-39250
START_RANGE client20a socket629 620 39250-39312
START_RANGE client20a socket630 630 39312-39375
START_RANGE client20a socket631 620 39375-39437
START_RANGE client20a socket632 630 39437-39500
START_RANGE client20a socket633 620 39500-39562
START_RANGE client20a socket634 630 39562-39625
START_RANGE client20a socket635 620 39625-39687
START_RANGE client20a socket636 630 39687-39750
START_RANGE client20a socket637 620 39750-39812
START_RANGE client20a socket638 630 39812-39875
START_RANGE client20a socket639 620 39875-39937
START_RANGE client20a socket640 630 39937-40000
#elif MASTER_NUM6
START_RANGE client21a socket641 620 40000-40062
START_RANGE client21a socket642 630 40062-40125
START_RANGE client21a socket643 620 40125-40187
START_RANGE client21a socket644 630 40187-40250
START_RANGE client21a socket645 620 40250-40312
START_RANGE client21a socket646 630 40312-40375
START_RANGE client21a socket647 620 40375-40437
START_RANGE client21a socket648 630 40437-40500
START_RANGE client21a socket649 620 40500-40562
START_RANGE client21a socket650 630 40562-40625
START_RANGE client21a socket651 620 40625-40687
START_RANGE client21a socket652 630 40687-40750
START_RANGE client21a socket653 620 40750-40812
START_RANGE client21a socket654 630 40812-40875
START_RANGE client21a socket655 620 40875-40937
START_RANGE client21a socket656 630 40937-41000
START_RANGE client21a socket657 620 41000-41062
START_RANGE client21a socket658 630 41062-41125
START_RANGE client21a socket659 620 41125-41187
START_RANGE client21a socket660 630 41187-41250
START_RANGE client21a socket661 620 41250-41312
START_RANGE client21a socket662 630 41312-41375
START_RANGE client21a socket663 620 41375-41437
START_RANGE client21a socket664 630 41437-41500
START_RANGE client21a socket665 620 41500-41562
START_RANGE client21a socket666 630 41562-41625
START_RANGE client21a socket667 620 41625-41687
START_RANGE client21a socket668 630 41687-41750
START_RANGE client21a socket669 620 41750-41812
START_RANGE client21a socket670 630 41812-41875
START_RANGE client21a socket671 620 41875-41937
START_RANGE client21a socket672 630 41937-42000
START_RANGE client22a socket673 620 42000-42062
START_RANGE client22a socket674 630 42062-42125
START_RANGE client22a socket675 620 42125-42187
START_RANGE client22a socket676 630 42187-42250
START_RANGE client22a socket677 620 42250-42312
START_RANGE client22a socket678 630 42312-42375
START_RANGE client22a socket679 620 42375-42437
START_RANGE client22a socket680 630 42437-42500
START_RANGE client22a socket681 620 42500-42562
START_RANGE client22a socket682 630 42562-42625
START_RANGE client22a socket683 620 42625-42687
START_RANGE client22a socket684 630 42687-42750
START_RANGE client22a socket685 620 42750-42812
START_RANGE client22a socket686 630 42812-42875

START_RANGE client22a socket687 620 42875-42937
START_RANGE client22a socket688 630 42937-43000
START_RANGE client22a socket689 620 43000-43062
START_RANGE client22a socket690 630 43062-43125
START_RANGE client22a socket691 620 43125-43187
START_RANGE client22a socket692 630 43187-43250
START_RANGE client22a socket693 620 43250-43312
START_RANGE client22a socket694 630 43312-43375
START_RANGE client22a socket695 620 43375-43437
START_RANGE client22a socket696 630 43437-43500
START_RANGE client22a socket697 620 43500-43562
START_RANGE client22a socket698 630 43562-43625
START_RANGE client22a socket699 620 43625-43687
START_RANGE client22a socket700 630 43687-43750
START_RANGE client22a socket701 620 43750-43812
START_RANGE client22a socket702 630 43812-43875
START_RANGE client22a socket703 620 43875-43937
START_RANGE client22a socket704 630 43937-44000
START_RANGE client23a socket705 620 44000-44062
START_RANGE client23a socket706 630 44062-44125
START_RANGE client23a socket707 620 44125-44187
START_RANGE client23a socket708 630 44187-44250
START_RANGE client23a socket709 620 44250-44312
START_RANGE client23a socket710 630 44312-44375
START_RANGE client23a socket711 620 44375-44437
START_RANGE client23a socket712 630 44437-44500
START_RANGE client23a socket713 620 44500-44562
START_RANGE client23a socket714 630 44562-44625
START_RANGE client23a socket715 620 44625-44687
START_RANGE client23a socket716 630 44687-44750
START_RANGE client23a socket717 620 44750-44812
START_RANGE client23a socket718 630 44812-44875
START_RANGE client23a socket719 620 44875-44937
START_RANGE client23a socket720 630 44937-45000
START_RANGE client23a socket721 620 45000-45062
START_RANGE client23a socket722 630 45062-45125
START_RANGE client23a socket723 620 45125-45187
START_RANGE client23a socket724 630 45187-45250
START_RANGE client23a socket725 620 45250-45312
START_RANGE client23a socket726 630 45312-45375
START_RANGE client23a socket727 620 45375-45437
START_RANGE client23a socket728 630 45437-45500
START_RANGE client23a socket729 620 45500-45562
START_RANGE client23a socket730 630 45562-45625
START_RANGE client23a socket731 620 45625-45687
START_RANGE client23a socket732 630 45687-45750
START_RANGE client23a socket733 620 45750-45812
START_RANGE client23a socket734 630 45812-45875
START_RANGE client23a socket735 620 45875-45937
START_RANGE client23a socket736 630 45937-46000
START_RANGE client24a socket737 620 46000-46062
START_RANGE client24a socket738 630 46062-46125
START_RANGE client24a socket739 620 46125-46187
START_RANGE client24a socket740 630 46187-46250
START_RANGE client24a socket741 620 46250-46312
START_RANGE client24a socket742 630 46312-46375
START_RANGE client24a socket743 620 46375-46437
START_RANGE client24a socket744 630 46437-46500
START_RANGE client24a socket745 620 46500-46562
START_RANGE client24a socket746 630 46562-46625
START_RANGE client24a socket747 620 46625-46687
START_RANGE client24a socket748 630 46687-46750

START_RANGE client24a socket749 620 46750-46812
START_RANGE client24a socket750 630 46812-46875
START_RANGE client24a socket751 620 46875-46937
START_RANGE client24a socket752 630 46937-47000
START_RANGE client24a socket753 620 47000-47062
START_RANGE client24a socket754 630 47062-47125
START_RANGE client24a socket755 620 47125-47187
START_RANGE client24a socket756 630 47187-47250
START_RANGE client24a socket757 620 47250-47312
START_RANGE client24a socket758 630 47312-47375
START_RANGE client24a socket759 620 47375-47437
START_RANGE client24a socket760 630 47437-47500
START_RANGE client24a socket761 620 47500-47562
START_RANGE client24a socket762 630 47562-47625
START_RANGE client24a socket763 620 47625-47687
START_RANGE client24a socket764 630 47687-47750
START_RANGE client24a socket765 620 47750-47812
START_RANGE client24a socket766 630 47812-47875
START_RANGE client24a socket767 620 47875-47937
START_RANGE client24a socket768 630 47937-48000
#elif MASTER_NUM7
START_RANGE client25a socket769 620 48000-48062
START_RANGE client25a socket770 630 48062-48125
START_RANGE client25a socket771 620 48125-48187
START_RANGE client25a socket772 630 48187-48250
START_RANGE client25a socket773 620 48250-48312
START_RANGE client25a socket774 630 48312-48375
START_RANGE client25a socket775 620 48375-48437
START_RANGE client25a socket776 630 48437-48500
START_RANGE client25a socket777 620 48500-48562
START_RANGE client25a socket778 630 48562-48625
START_RANGE client25a socket779 620 48625-48687
START_RANGE client25a socket780 630 48687-48750
START_RANGE client25a socket781 620 48750-48812
START_RANGE client25a socket782 630 48812-48875
START_RANGE client25a socket783 620 48875-48937
START_RANGE client25a socket784 630 48937-49000
START_RANGE client25a socket785 620 49000-49062
START_RANGE client25a socket786 630 49062-49125
START_RANGE client25a socket787 620 49125-49187
START_RANGE client25a socket788 630 49187-49250
START_RANGE client25a socket789 620 49250-49312
START_RANGE client25a socket790 630 49312-49375
START_RANGE client25a socket791 620 49375-49437
START_RANGE client25a socket792 630 49437-49500
START_RANGE client25a socket793 620 49500-49562
START_RANGE client25a socket794 630 49562-49625
START_RANGE client25a socket795 620 49625-49687
START_RANGE client25a socket796 630 49687-49750
START_RANGE client25a socket797 620 49750-49812
START_RANGE client25a socket798 630 49812-49875
START_RANGE client25a socket799 620 49875-49937
START_RANGE client25a socket800 630 49937-50000
START_RANGE client26a socket801 620 50000-50062
START_RANGE client26a socket802 630 50062-50125
START_RANGE client26a socket803 620 50125-50187
START_RANGE client26a socket804 630 50187-50250
START_RANGE client26a socket805 620 50250-50312
START_RANGE client26a socket806 630 50312-50375
START_RANGE client26a socket807 620 50375-50437
START_RANGE client26a socket808 630 50437-50500
START_RANGE client26a socket809 620 50500-50562

START_RANGE client26a socket810 630 50562-50625
START_RANGE client26a socket811 620 50625-50687
START_RANGE client26a socket812 630 50687-50750
START_RANGE client26a socket813 620 50750-50812
START_RANGE client26a socket814 630 50812-50875
START_RANGE client26a socket815 620 50875-50937
START_RANGE client26a socket816 630 50937-51000
START_RANGE client26a socket817 620 51000-51062
START_RANGE client26a socket818 630 51062-51125
START_RANGE client26a socket819 620 51125-51187
START_RANGE client26a socket820 630 51187-51250
START_RANGE client26a socket821 620 51250-51312
START_RANGE client26a socket822 630 51312-51375
START_RANGE client26a socket823 620 51375-51437
START_RANGE client26a socket824 630 51437-51500
START_RANGE client26a socket825 620 51500-51562
START_RANGE client26a socket826 630 51562-51625
START_RANGE client26a socket827 620 51625-51687
START_RANGE client26a socket828 630 51687-51750
START_RANGE client26a socket829 620 51750-51812
START_RANGE client26a socket830 630 51812-51875
START_RANGE client26a socket831 620 51875-51937
START_RANGE client26a socket832 630 51937-52000
START_RANGE client27a socket833 620 52000-52062
START_RANGE client27a socket834 630 52062-52125
START_RANGE client27a socket835 620 52125-52187
START_RANGE client27a socket836 630 52187-52250
START_RANGE client27a socket837 620 52250-52312
START_RANGE client27a socket838 630 52312-52375
START_RANGE client27a socket839 620 52375-52437
START_RANGE client27a socket840 630 52437-52500
START_RANGE client27a socket841 620 52500-52562
START_RANGE client27a socket842 630 52562-52625
START_RANGE client27a socket843 620 52625-52687
START_RANGE client27a socket844 630 52687-52750
START_RANGE client27a socket845 620 52750-52812
START_RANGE client27a socket846 630 52812-52875
START_RANGE client27a socket847 620 52875-52937
START_RANGE client27a socket848 630 52937-53000
START_RANGE client27a socket849 620 53000-53062
START_RANGE client27a socket850 630 53062-53125
START_RANGE client27a socket851 620 53125-53187
START_RANGE client27a socket852 630 53187-53250
START_RANGE client27a socket853 620 53250-53312
START_RANGE client27a socket854 630 53312-53375
START_RANGE client27a socket855 620 53375-53437
START_RANGE client27a socket856 630 53437-53500
START_RANGE client27a socket857 620 53500-53562
START_RANGE client27a socket858 630 53562-53625
START_RANGE client27a socket859 620 53625-53687
START_RANGE client27a socket860 630 53687-53750
START_RANGE client27a socket861 620 53750-53812
START_RANGE client27a socket862 630 53812-53875
START_RANGE client27a socket863 620 53875-53937
START_RANGE client27a socket864 630 53937-54000
START_RANGE client28a socket865 620 54000-54062
START_RANGE client28a socket866 630 54062-54125
START_RANGE client28a socket867 620 54125-54187
START_RANGE client28a socket868 630 54187-54250
START_RANGE client28a socket869 620 54250-54312
START_RANGE client28a socket870 630 54312-54375
START_RANGE client28a socket871 620 54375-54437

START_RANGE client28a socket872 630 54437-54500
START_RANGE client28a socket873 620 54500-54562
START_RANGE client28a socket874 630 54562-54625
START_RANGE client28a socket875 620 54625-54687
START_RANGE client28a socket876 630 54687-54750
START_RANGE client28a socket877 620 54750-54812
START_RANGE client28a socket878 630 54812-54875
START_RANGE client28a socket879 620 54875-54937
START_RANGE client28a socket880 630 54937-55000
START_RANGE client28a socket881 620 55000-55062
START_RANGE client28a socket882 630 55062-55125
START_RANGE client28a socket883 620 55125-55187
START_RANGE client28a socket884 630 55187-55250
START_RANGE client28a socket885 620 55250-55312
START_RANGE client28a socket886 630 55312-55375
START_RANGE client28a socket887 620 55375-55437
START_RANGE client28a socket888 630 55437-55500
START_RANGE client28a socket889 620 55500-55562
START_RANGE client28a socket890 630 55562-55625
START_RANGE client28a socket891 620 55625-55687
START_RANGE client28a socket892 630 55687-55750
START_RANGE client28a socket893 620 55750-55812
START_RANGE client28a socket894 630 55812-55875
START_RANGE client28a socket895 620 55875-55937
START_RANGE client28a socket896 630 55937-56000
#elif MASTER_NUM8
START_RANGE client29a socket897 620 56000-56062
START_RANGE client29a socket898 630 56062-56125
START_RANGE client29a socket899 620 56125-56187
START_RANGE client29a socket900 630 56187-56250
START_RANGE client29a socket901 620 56250-56312
START_RANGE client29a socket902 630 56312-56375
START_RANGE client29a socket903 620 56375-56437
START_RANGE client29a socket904 630 56437-56500
START_RANGE client29a socket905 620 56500-56562
START_RANGE client29a socket906 630 56562-56625
START_RANGE client29a socket907 620 56625-56687
START_RANGE client29a socket908 630 56687-56750
START_RANGE client29a socket909 620 56750-56812
START_RANGE client29a socket910 630 56812-56875
START_RANGE client29a socket911 620 56875-56937
START_RANGE client29a socket912 630 56937-57000
START_RANGE client29a socket913 620 57000-57062
START_RANGE client29a socket914 630 57062-57125
START_RANGE client29a socket915 620 57125-57187
START_RANGE client29a socket916 630 57187-57250
START_RANGE client29a socket917 620 57250-57312
START_RANGE client29a socket918 630 57312-57375
START_RANGE client29a socket919 620 57375-57437
START_RANGE client29a socket920 630 57437-57500
START_RANGE client29a socket921 620 57500-57562
START_RANGE client29a socket922 630 57562-57625
START_RANGE client29a socket923 620 57625-57687
START_RANGE client29a socket924 630 57687-57750
START_RANGE client29a socket925 620 57750-57812
START_RANGE client29a socket926 630 57812-57875
START_RANGE client29a socket927 620 57875-57937
START_RANGE client29a socket928 630 57937-58000
START_RANGE client30a socket929 620 58000-58062
START_RANGE client30a socket930 630 58062-58125
START_RANGE client30a socket931 620 58125-58187
START_RANGE client30a socket932 630 58187-58250

START_RANGE client30a socket933 620 58250-58312
START_RANGE client30a socket934 630 58312-58375
START_RANGE client30a socket935 620 58375-58437
START_RANGE client30a socket936 630 58437-58500
START_RANGE client30a socket937 620 58500-58562
START_RANGE client30a socket938 630 58562-58625
START_RANGE client30a socket939 620 58625-58687
START_RANGE client30a socket940 630 58687-58750
START_RANGE client30a socket941 620 58750-58812
START_RANGE client30a socket942 630 58812-58875
START_RANGE client30a socket943 620 58875-58937
START_RANGE client30a socket944 630 58937-59000
START_RANGE client30a socket945 620 59000-59062
START_RANGE client30a socket946 630 59062-59125
START_RANGE client30a socket947 620 59125-59187
START_RANGE client30a socket948 630 59187-59250
START_RANGE client30a socket949 620 59250-59312
START_RANGE client30a socket950 630 59312-59375
START_RANGE client30a socket951 620 59375-59437
START_RANGE client30a socket952 630 59437-59500
START_RANGE client30a socket953 620 59500-59562
START_RANGE client30a socket954 630 59562-59625
START_RANGE client30a socket955 620 59625-59687
START_RANGE client30a socket956 630 59687-59750
START_RANGE client30a socket957 620 59750-59812
START_RANGE client30a socket958 630 59812-59875
START_RANGE client30a socket959 620 59875-59937
START_RANGE client30a socket960 630 59937-60000
START_RANGE client31a socket961 620 60000-60062
START_RANGE client31a socket962 630 60062-60125
START_RANGE client31a socket963 620 60125-60187
START_RANGE client31a socket964 630 60187-60250
START_RANGE client31a socket965 620 60250-60312
START_RANGE client31a socket966 630 60312-60375
START_RANGE client31a socket967 620 60375-60437
START_RANGE client31a socket968 630 60437-60500
START_RANGE client31a socket969 620 60500-60562
START_RANGE client31a socket970 630 60562-60625
START_RANGE client31a socket971 620 60625-60687
START_RANGE client31a socket972 630 60687-60750
START_RANGE client31a socket973 620 60750-60812
START_RANGE client31a socket974 630 60812-60875
START_RANGE client31a socket975 620 60875-60937
START_RANGE client31a socket976 630 60937-61000
START_RANGE client31a socket977 620 61000-61062
START_RANGE client31a socket978 630 61062-61125
START_RANGE client31a socket979 620 61125-61187
START_RANGE client31a socket980 630 61187-61250
START_RANGE client31a socket981 620 61250-61312
START_RANGE client31a socket982 630 61312-61375
START_RANGE client31a socket983 620 61375-61437
START_RANGE client31a socket984 630 61437-61500
START_RANGE client31a socket985 620 61500-61562
START_RANGE client31a socket986 630 61562-61625
START_RANGE client31a socket987 620 61625-61687
START_RANGE client31a socket988 630 61687-61750
START_RANGE client31a socket989 620 61750-61812
START_RANGE client31a socket990 630 61812-61875
START_RANGE client31a socket991 620 61875-61937
START_RANGE client31a socket992 630 61937-62000
START_RANGE client32a socket993 620 62000-62062
START_RANGE client32a socket994 630 62062-62125

START_RANGE client32a socket995 620 62125-62187
START_RANGE client32a socket996 630 62187-62250
START_RANGE client32a socket997 620 62250-62312
START_RANGE client32a socket998 630 62312-62375
START_RANGE client32a socket999 620 62375-62437
START_RANGE client32a socket1000 630 62437-62500
START_RANGE client32a socket1001 620 62500-62562
START_RANGE client32a socket1002 630 62562-62625
START_RANGE client32a socket1003 620 62625-62687
START_RANGE client32a socket1004 630 62687-62750
START_RANGE client32a socket1005 620 62750-62812
START_RANGE client32a socket1006 630 62812-62875
START_RANGE client32a socket1007 620 62875-62937
START_RANGE client32a socket1008 630 62937-63000
START_RANGE client32a socket1009 620 63000-63062
START_RANGE client32a socket1010 630 63062-63125
START_RANGE client32a socket1011 620 63125-63187
START_RANGE client32a socket1012 630 63187-63250
START_RANGE client32a socket1013 620 63250-63312
START_RANGE client32a socket1014 630 63312-63375
START_RANGE client32a socket1015 620 63375-63437
START_RANGE client32a socket1016 630 63437-63500
START_RANGE client32a socket1017 620 63500-63562
START_RANGE client32a socket1018 630 63562-63625
START_RANGE client32a socket1019 620 63625-63687
START_RANGE client32a socket1020 630 63687-63750
START_RANGE client32a socket1021 620 63750-63812
START_RANGE client32a socket1022 630 63812-63875
START_RANGE client32a socket1023 620 63875-63937
START_RANGE client32a socket1024 630 63937-64000
#elif MASTER_NUM9
START_RANGE client33a socket1025 620 64000-64062
START_RANGE client33a socket1026 630 64062-64125
START_RANGE client33a socket1027 620 64125-64187
START_RANGE client33a socket1028 630 64187-64250
START_RANGE client33a socket1029 620 64250-64312
START_RANGE client33a socket1030 630 64312-64375
START_RANGE client33a socket1031 620 64375-64437
START_RANGE client33a socket1032 630 64437-64500
START_RANGE client33a socket1033 620 64500-64562
START_RANGE client33a socket1034 630 64562-64625
START_RANGE client33a socket1035 620 64625-64687
START_RANGE client33a socket1036 630 64687-64750
START_RANGE client33a socket1037 620 64750-64812
START_RANGE client33a socket1038 630 64812-64875
START_RANGE client33a socket1039 620 64875-64937
START_RANGE client33a socket1040 630 64937-65000
START_RANGE client33a socket1041 620 65000-65062
START_RANGE client33a socket1042 630 65062-65125
START_RANGE client33a socket1043 620 65125-65187
START_RANGE client33a socket1044 630 65187-65250
START_RANGE client33a socket1045 620 65250-65312
START_RANGE client33a socket1046 630 65312-65375
START_RANGE client33a socket1047 620 65375-65437
START_RANGE client33a socket1048 630 65437-65500
START_RANGE client33a socket1049 620 65500-65562
START_RANGE client33a socket1050 630 65562-65625
START_RANGE client33a socket1051 620 65625-65687
START_RANGE client33a socket1052 630 65687-65750
START_RANGE client33a socket1053 620 65750-65812
START_RANGE client33a socket1054 630 65812-65875
START_RANGE client33a socket1055 620 65875-65937

START_RANGE client33a socket1056 630 65937-66000
START_RANGE client34a socket1057 620 66000-66062
START_RANGE client34a socket1058 630 66062-66125
START_RANGE client34a socket1059 620 66125-66187
START_RANGE client34a socket1060 630 66187-66250
START_RANGE client34a socket1061 620 66250-66312
START_RANGE client34a socket1062 630 66312-66375
START_RANGE client34a socket1063 620 66375-66437
START_RANGE client34a socket1064 630 66437-66500
START_RANGE client34a socket1065 620 66500-66562
START_RANGE client34a socket1066 630 66562-66625
START_RANGE client34a socket1067 620 66625-66687
START_RANGE client34a socket1068 630 66687-66750
START_RANGE client34a socket1069 620 66750-66812
START_RANGE client34a socket1070 630 66812-66875
START_RANGE client34a socket1071 620 66875-66937
START_RANGE client34a socket1072 630 66937-67000
START_RANGE client34a socket1073 620 67000-67062
START_RANGE client34a socket1074 630 67062-67125
START_RANGE client34a socket1075 620 67125-67187
START_RANGE client34a socket1076 630 67187-67250
START_RANGE client34a socket1077 620 67250-67312
START_RANGE client34a socket1078 630 67312-67375
START_RANGE client34a socket1079 620 67375-67437
START_RANGE client34a socket1080 630 67437-67500
START_RANGE client34a socket1081 620 67500-67562
START_RANGE client34a socket1082 630 67562-67625
START_RANGE client34a socket1083 620 67625-67687
START_RANGE client34a socket1084 630 67687-67750
START_RANGE client34a socket1085 620 67750-67812
START_RANGE client34a socket1086 630 67812-67875
START_RANGE client34a socket1087 620 67875-67937
START_RANGE client34a socket1088 630 67937-68000
START_RANGE client35a socket1089 620 68000-68062
START_RANGE client35a socket1090 630 68062-68125
START_RANGE client35a socket1091 620 68125-68187
START_RANGE client35a socket1092 630 68187-68250
START_RANGE client35a socket1093 620 68250-68312
START_RANGE client35a socket1094 630 68312-68375
START_RANGE client35a socket1095 620 68375-68437
START_RANGE client35a socket1096 630 68437-68500
START_RANGE client35a socket1097 620 68500-68562
START_RANGE client35a socket1098 630 68562-68625
START_RANGE client35a socket1099 620 68625-68687
START_RANGE client35a socket1100 630 68687-68750
START_RANGE client35a socket1101 620 68750-68812
START_RANGE client35a socket1102 630 68812-68875
START_RANGE client35a socket1103 620 68875-68937
START_RANGE client35a socket1104 630 68937-69000
START_RANGE client35a socket1105 620 69000-69062
START_RANGE client35a socket1106 630 69062-69125
START_RANGE client35a socket1107 620 69125-69187
START_RANGE client35a socket1108 630 69187-69250
START_RANGE client35a socket1109 620 69250-69312
START_RANGE client35a socket1110 630 69312-69375
START_RANGE client35a socket1111 620 69375-69437
START_RANGE client35a socket1112 630 69437-69500
START_RANGE client35a socket1113 620 69500-69562
START_RANGE client35a socket1114 630 69562-69625
START_RANGE client35a socket1115 620 69625-69687
START_RANGE client35a socket1116 630 69687-69750
START_RANGE client35a socket1117 620 69750-69812

START_RANGE client35a socket1118 630 69812-69875
START_RANGE client35a socket1119 620 69875-69937
START_RANGE client35a socket1120 630 69937-70000
START_RANGE client36a socket1121 620 70000-70062
START_RANGE client36a socket1122 630 70062-70125
START_RANGE client36a socket1123 620 70125-70187
START_RANGE client36a socket1124 630 70187-70250
START_RANGE client36a socket1125 620 70250-70312
START_RANGE client36a socket1126 630 70312-70375
START_RANGE client36a socket1127 620 70375-70437
START_RANGE client36a socket1128 630 70437-70500
START_RANGE client36a socket1129 620 70500-70562
START_RANGE client36a socket1130 630 70562-70625
START_RANGE client36a socket1131 620 70625-70687
START_RANGE client36a socket1132 630 70687-70750
START_RANGE client36a socket1133 620 70750-70812
START_RANGE client36a socket1134 630 70812-70875
START_RANGE client36a socket1135 620 70875-70937
START_RANGE client36a socket1136 630 70937-71000
START_RANGE client36a socket1137 620 71000-71062
START_RANGE client36a socket1138 630 71062-71125
START_RANGE client36a socket1139 620 71125-71187
START_RANGE client36a socket1140 630 71187-71250
START_RANGE client36a socket1141 620 71250-71312
START_RANGE client36a socket1142 630 71312-71375
START_RANGE client36a socket1143 620 71375-71437
START_RANGE client36a socket1144 630 71437-71500
START_RANGE client36a socket1145 620 71500-71562
START_RANGE client36a socket1146 630 71562-71625
START_RANGE client36a socket1147 620 71625-71687
START_RANGE client36a socket1148 630 71687-71750
START_RANGE client36a socket1149 620 71750-71812
START_RANGE client36a socket1150 630 71812-71875
START_RANGE client36a socket1151 620 71875-71937
START_RANGE client36a socket1152 630 71937-72000
#elif MASTER_NUM10
START_RANGE client37a socket1153 620 72000-72062
START_RANGE client37a socket1154 630 72062-72125
START_RANGE client37a socket1155 620 72125-72187
START_RANGE client37a socket1156 630 72187-72250
START_RANGE client37a socket1157 620 72250-72312
START_RANGE client37a socket1158 630 72312-72375
START_RANGE client37a socket1159 620 72375-72437
START_RANGE client37a socket1160 630 72437-72500
START_RANGE client37a socket1161 620 72500-72562
START_RANGE client37a socket1162 630 72562-72625
START_RANGE client37a socket1163 620 72625-72687
START_RANGE client37a socket1164 630 72687-72750
START_RANGE client37a socket1165 620 72750-72812
START_RANGE client37a socket1166 630 72812-72875
START_RANGE client37a socket1167 620 72875-72937
START_RANGE client37a socket1168 630 72937-73000
START_RANGE client37a socket1169 620 73000-73062
START_RANGE client37a socket1170 630 73062-73125
START_RANGE client37a socket1171 620 73125-73187
START_RANGE client37a socket1172 630 73187-73250
START_RANGE client37a socket1173 620 73250-73312
START_RANGE client37a socket1174 630 73312-73375
START_RANGE client37a socket1175 620 73375-73437
START_RANGE client37a socket1176 630 73437-73500
START_RANGE client37a socket1177 620 73500-73562
START_RANGE client37a socket1178 630 73562-73625

START_RANGE client37a socket1179 620 73625-73687
START_RANGE client37a socket1180 630 73687-73750
START_RANGE client37a socket1181 620 73750-73812
START_RANGE client37a socket1182 630 73812-73875
START_RANGE client37a socket1183 620 73875-73937
START_RANGE client37a socket1184 630 73937-74000
START_RANGE client38a socket1185 620 74000-74062
START_RANGE client38a socket1186 630 74062-74125
START_RANGE client38a socket1187 620 74125-74187
START_RANGE client38a socket1188 630 74187-74250
START_RANGE client38a socket1189 620 74250-74312
START_RANGE client38a socket1190 630 74312-74375
START_RANGE client38a socket1191 620 74375-74437
START_RANGE client38a socket1192 630 74437-74500
START_RANGE client38a socket1193 620 74500-74562
START_RANGE client38a socket1194 630 74562-74625
START_RANGE client38a socket1195 620 74625-74687
START_RANGE client38a socket1196 630 74687-74750
START_RANGE client38a socket1197 620 74750-74812
START_RANGE client38a socket1198 630 74812-74875
START_RANGE client38a socket1199 620 74875-74937
START_RANGE client38a socket1200 630 74937-75000
START_RANGE client38a socket1201 620 75000-75062
START_RANGE client38a socket1202 630 75062-75125
START_RANGE client38a socket1203 620 75125-75187
START_RANGE client38a socket1204 630 75187-75250
START_RANGE client38a socket1205 620 75250-75312
START_RANGE client38a socket1206 630 75312-75375
START_RANGE client38a socket1207 620 75375-75437
START_RANGE client38a socket1208 630 75437-75500
START_RANGE client38a socket1209 620 75500-75562
START_RANGE client38a socket1210 630 75562-75625
START_RANGE client38a socket1211 620 75625-75687
START_RANGE client38a socket1212 630 75687-75750
START_RANGE client38a socket1213 620 75750-75812
START_RANGE client38a socket1214 630 75812-75875
START_RANGE client38a socket1215 620 75875-75937
START_RANGE client38a socket1216 630 75937-76000
START_RANGE client39a socket1217 620 76000-76062
START_RANGE client39a socket1218 630 76062-76125
START_RANGE client39a socket1219 620 76125-76187
START_RANGE client39a socket1220 630 76187-76250
START_RANGE client39a socket1221 620 76250-76312
START_RANGE client39a socket1222 630 76312-76375
START_RANGE client39a socket1223 620 76375-76437
START_RANGE client39a socket1224 630 76437-76500
START_RANGE client39a socket1225 620 76500-76562
START_RANGE client39a socket1226 630 76562-76625
START_RANGE client39a socket1227 620 76625-76687
START_RANGE client39a socket1228 630 76687-76750
START_RANGE client39a socket1229 620 76750-76812
START_RANGE client39a socket1230 630 76812-76875
START_RANGE client39a socket1231 620 76875-76937
START_RANGE client39a socket1232 630 76937-77000
START_RANGE client39a socket1233 620 77000-77062
START_RANGE client39a socket1234 630 77062-77125
START_RANGE client39a socket1235 620 77125-77187
START_RANGE client39a socket1236 630 77187-77250
START_RANGE client39a socket1237 620 77250-77312
START_RANGE client39a socket1238 630 77312-77375
START_RANGE client39a socket1239 620 77375-77437
START_RANGE client39a socket1240 630 77437-77500

START_RANGE client39a socket1241 620 77500-77562
START_RANGE client39a socket1242 630 77562-77625
START_RANGE client39a socket1243 620 77625-77687
START_RANGE client39a socket1244 630 77687-77750
START_RANGE client39a socket1245 620 77750-77812
START_RANGE client39a socket1246 630 77812-77875
START_RANGE client39a socket1247 620 77875-77937
START_RANGE client39a socket1248 630 77937-78000
START_RANGE client40a socket1249 620 78000-78062
START_RANGE client40a socket1250 630 78062-78125
START_RANGE client40a socket1251 620 78125-78187
START_RANGE client40a socket1252 630 78187-78250
START_RANGE client40a socket1253 620 78250-78312
START_RANGE client40a socket1254 630 78312-78375
START_RANGE client40a socket1255 620 78375-78437
START_RANGE client40a socket1256 630 78437-78500
START_RANGE client40a socket1257 620 78500-78562
START_RANGE client40a socket1258 630 78562-78625
START_RANGE client40a socket1259 620 78625-78687
START_RANGE client40a socket1260 630 78687-78750
START_RANGE client40a socket1261 620 78750-78812
START_RANGE client40a socket1262 630 78812-78875
START_RANGE client40a socket1263 620 78875-78937
START_RANGE client40a socket1264 630 78937-79000
START_RANGE client40a socket1265 620 79000-79062
START_RANGE client40a socket1266 630 79062-79125
START_RANGE client40a socket1267 620 79125-79187
START_RANGE client40a socket1268 630 79187-79250
START_RANGE client40a socket1269 620 79250-79312
START_RANGE client40a socket1270 630 79312-79375
START_RANGE client40a socket1271 620 79375-79437
START_RANGE client40a socket1272 630 79437-79500
START_RANGE client40a socket1273 620 79500-79562
START_RANGE client40a socket1274 630 79562-79625
START_RANGE client40a socket1275 620 79625-79687
START_RANGE client40a socket1276 630 79687-79750
START_RANGE client40a socket1277 620 79750-79812
START_RANGE client40a socket1278 630 79812-79875
START_RANGE client40a socket1279 620 79875-79937
START_RANGE client40a socket1280 630 79937-80000
#elif MASTER_NUM11
START_RANGE client41a socket1281 620 80000-80062
START_RANGE client41a socket1282 630 80062-80125
START_RANGE client41a socket1283 620 80125-80187
START_RANGE client41a socket1284 630 80187-80250
START_RANGE client41a socket1285 620 80250-80312
START_RANGE client41a socket1286 630 80312-80375
START_RANGE client41a socket1287 620 80375-80437
START_RANGE client41a socket1288 630 80437-80500
START_RANGE client41a socket1289 620 80500-80562
START_RANGE client41a socket1290 630 80562-80625
START_RANGE client41a socket1291 620 80625-80687
START_RANGE client41a socket1292 630 80687-80750
START_RANGE client41a socket1293 620 80750-80812
START_RANGE client41a socket1294 630 80812-80875
START_RANGE client41a socket1295 620 80875-80937
START_RANGE client41a socket1296 630 80937-81000
START_RANGE client41a socket1297 620 81000-81062
START_RANGE client41a socket1298 630 81062-81125
START_RANGE client41a socket1299 620 81125-81187
START_RANGE client41a socket1300 630 81187-81250
START_RANGE client41a socket1301 620 81250-81312

START_RANGE client41a socket1302 630 81312-81375
START_RANGE client41a socket1303 620 81375-81437
START_RANGE client41a socket1304 630 81437-81500
START_RANGE client41a socket1305 620 81500-81562
START_RANGE client41a socket1306 630 81562-81625
START_RANGE client41a socket1307 620 81625-81687
START_RANGE client41a socket1308 630 81687-81750
START_RANGE client41a socket1309 620 81750-81812
START_RANGE client41a socket1310 630 81812-81875
START_RANGE client41a socket1311 620 81875-81937
START_RANGE client41a socket1312 630 81937-82000
START_RANGE client42a socket1313 620 82000-82062
START_RANGE client42a socket1314 630 82062-82125
START_RANGE client42a socket1315 620 82125-82187
START_RANGE client42a socket1316 630 82187-82250
START_RANGE client42a socket1317 620 82250-82312
START_RANGE client42a socket1318 630 82312-82375
START_RANGE client42a socket1319 620 82375-82437
START_RANGE client42a socket1320 630 82437-82500
START_RANGE client42a socket1321 620 82500-82562
START_RANGE client42a socket1322 630 82562-82625
START_RANGE client42a socket1323 620 82625-82687
START_RANGE client42a socket1324 630 82687-82750
START_RANGE client42a socket1325 620 82750-82812
START_RANGE client42a socket1326 630 82812-82875
START_RANGE client42a socket1327 620 82875-82937
START_RANGE client42a socket1328 630 82937-83000
START_RANGE client42a socket1329 620 83000-83062
START_RANGE client42a socket1330 630 83062-83125
START_RANGE client42a socket1331 620 83125-83187
START_RANGE client42a socket1332 630 83187-83250
START_RANGE client42a socket1333 620 83250-83312
START_RANGE client42a socket1334 630 83312-83375
START_RANGE client42a socket1335 620 83375-83437
START_RANGE client42a socket1336 630 83437-83500
START_RANGE client42a socket1337 620 83500-83562
START_RANGE client42a socket1338 630 83562-83625
START_RANGE client42a socket1339 620 83625-83687
START_RANGE client42a socket1340 630 83687-83750
START_RANGE client42a socket1341 620 83750-83812
START_RANGE client42a socket1342 630 83812-83875
START_RANGE client42a socket1343 620 83875-83937
START_RANGE client42a socket1344 630 83937-84000
START_RANGE client42a socket1345 620 84000-84062
START_RANGE client43a socket1346 630 84062-84125
START_RANGE client43a socket1347 620 84125-84187
START_RANGE client43a socket1348 630 84187-84250
START_RANGE client43a socket1349 620 84250-84312
START_RANGE client43a socket1350 630 84312-84375
START_RANGE client43a socket1351 620 84375-84437
START_RANGE client43a socket1352 630 84437-84500
START_RANGE client43a socket1353 620 84500-84562
START_RANGE client43a socket1354 630 84562-84625
START_RANGE client43a socket1355 620 84625-84687
START_RANGE client43a socket1356 630 84687-84750
START_RANGE client43a socket1357 620 84750-84812
START_RANGE client43a socket1358 630 84812-84875
START_RANGE client43a socket1359 620 84875-84937
START_RANGE client43a socket1360 630 84937-85000
START_RANGE client43a socket1361 620 85000-85062
START_RANGE client43a socket1362 630 85062-85125
START_RANGE client43a socket1363 620 85125-85187

START_RANGE client43a socket1364 630 85187-85250
START_RANGE client43a socket1365 620 85250-85312
START_RANGE client43a socket1366 630 85312-85375
START_RANGE client43a socket1367 620 85375-85437
START_RANGE client43a socket1368 630 85437-85500
START_RANGE client43a socket1369 620 85500-85562
START_RANGE client43a socket1370 630 85562-85625
START_RANGE client43a socket1371 620 85625-85687
START_RANGE client43a socket1372 630 85687-85750
START_RANGE client43a socket1373 620 85750-85812
START_RANGE client43a socket1374 630 85812-85875
START_RANGE client43a socket1375 620 85875-85937
START_RANGE client43a socket1376 630 85937-86000
START_RANGE client44a socket1377 620 86000-86062
START_RANGE client44a socket1378 630 86062-86125
START_RANGE client44a socket1379 620 86125-86187
START_RANGE client44a socket1380 630 86187-86250
START_RANGE client44a socket1381 620 86250-86312
START_RANGE client44a socket1382 630 86312-86375
START_RANGE client44a socket1383 620 86375-86437
START_RANGE client44a socket1384 630 86437-86500
START_RANGE client44a socket1385 620 86500-86562
START_RANGE client44a socket1386 630 86562-86625
START_RANGE client44a socket1387 620 86625-86687
START_RANGE client44a socket1388 630 86687-86750
START_RANGE client44a socket1389 620 86750-86812
START_RANGE client44a socket1390 630 86812-86875
START_RANGE client44a socket1391 620 86875-86937
START_RANGE client44a socket1392 630 86937-87000
START_RANGE client44a socket1393 620 87000-87062
START_RANGE client44a socket1394 630 87062-87125
START_RANGE client44a socket1395 620 87125-87187
START_RANGE client44a socket1396 630 87187-87250
START_RANGE client44a socket1397 620 87250-87312
START_RANGE client44a socket1398 630 87312-87375
START_RANGE client44a socket1399 620 87375-87437
START_RANGE client44a socket1400 630 87437-87500
START_RANGE client44a socket1401 620 87500-87562
START_RANGE client44a socket1402 630 87562-87625
START_RANGE client44a socket1403 620 87625-87687
START_RANGE client44a socket1404 630 87687-87750
START_RANGE client44a socket1405 620 87750-87812
START_RANGE client44a socket1406 630 87812-87875
START_RANGE client44a socket1407 620 87875-87937
START_RANGE client44a socket1408 630 87937-88000
#elif MASTER_NUM12
START_RANGE client45a socket1409 620 88000-88062
START_RANGE client45a socket1410 630 88062-88125
START_RANGE client45a socket1411 620 88125-88187
START_RANGE client45a socket1412 630 88187-88250
START_RANGE client45a socket1413 620 88250-88312
START_RANGE client45a socket1414 630 88312-88375
START_RANGE client45a socket1415 620 88375-88437
START_RANGE client45a socket1416 630 88437-88500
START_RANGE client45a socket1417 620 88500-88562
START_RANGE client45a socket1418 630 88562-88625
START_RANGE client45a socket1419 620 88625-88687
START_RANGE client45a socket1420 630 88687-88750
START_RANGE client45a socket1421 620 88750-88812
START_RANGE client45a socket1422 630 88812-88875
START_RANGE client45a socket1423 620 88875-88937
START_RANGE client45a socket1424 630 88937-89000

START_RANGE client45a socket1425 620 89000-89062
START_RANGE client45a socket1426 630 89062-89125
START_RANGE client45a socket1427 620 89125-89187
START_RANGE client45a socket1428 630 89187-89250
START_RANGE client45a socket1429 620 89250-89312
START_RANGE client45a socket1430 630 89312-89375
START_RANGE client45a socket1431 620 89375-89437
START_RANGE client45a socket1432 630 89437-89500
START_RANGE client45a socket1433 620 89500-89562
START_RANGE client45a socket1434 630 89562-89625
START_RANGE client45a socket1435 620 89625-89687
START_RANGE client45a socket1436 630 89687-89750
START_RANGE client45a socket1437 620 89750-89812
START_RANGE client45a socket1438 630 89812-89875
START_RANGE client45a socket1439 620 89875-89937
START_RANGE client45a socket1440 630 89937-90000
START_RANGE client46a socket1441 620 90000-90062
START_RANGE client46a socket1442 630 90062-90125
START_RANGE client46a socket1443 620 90125-90187
START_RANGE client46a socket1444 630 90187-90250
START_RANGE client46a socket1445 620 90250-90312
START_RANGE client46a socket1446 630 90312-90375
START_RANGE client46a socket1447 620 90375-90437
START_RANGE client46a socket1448 630 90437-90500
START_RANGE client46a socket1449 620 90500-90562
START_RANGE client46a socket1450 630 90562-90625
START_RANGE client46a socket1451 620 90625-90687
START_RANGE client46a socket1452 630 90687-90750
START_RANGE client46a socket1453 620 90750-90812
START_RANGE client46a socket1454 630 90812-90875
START_RANGE client46a socket1455 620 90875-90937
START_RANGE client46a socket1456 630 90937-91000
START_RANGE client46a socket1457 620 91000-91062
START_RANGE client46a socket1458 630 91062-91125
START_RANGE client46a socket1459 620 91125-91187
START_RANGE client46a socket1460 630 91187-91250
START_RANGE client46a socket1461 620 91250-91312
START_RANGE client46a socket1462 630 91312-91375
START_RANGE client46a socket1463 620 91375-91437
START_RANGE client46a socket1464 630 91437-91500
START_RANGE client46a socket1465 620 91500-91562
START_RANGE client46a socket1466 630 91562-91625
START_RANGE client46a socket1467 620 91625-91687
START_RANGE client46a socket1468 630 91687-91750
START_RANGE client46a socket1469 620 91750-91812
START_RANGE client46a socket1470 630 91812-91875
START_RANGE client46a socket1471 620 91875-91937
START_RANGE client46a socket1472 630 91937-92000
START_RANGE client47a socket1473 620 92000-92062
START_RANGE client47a socket1474 630 92062-92125
START_RANGE client47a socket1475 620 92125-92187
START_RANGE client47a socket1476 630 92187-92250
START_RANGE client47a socket1477 620 92250-92312
START_RANGE client47a socket1478 630 92312-92375
START_RANGE client47a socket1479 620 92375-92437
START_RANGE client47a socket1480 630 92437-92500
START_RANGE client47a socket1481 620 92500-92562
START_RANGE client47a socket1482 630 92562-92625
START_RANGE client47a socket1483 620 92625-92687
START_RANGE client47a socket1484 630 92687-92750
START_RANGE client47a socket1485 620 92750-92812
START_RANGE client47a socket1486 630 92812-92875

START_RANGE client47a socket1487 620 92875-92937
START_RANGE client47a socket1488 630 92937-93000
START_RANGE client47a socket1489 620 93000-93062
START_RANGE client47a socket1490 630 93062-93125
START_RANGE client47a socket1491 620 93125-93187
START_RANGE client47a socket1492 630 93187-93250
START_RANGE client47a socket1493 620 93250-93312
START_RANGE client47a socket1494 630 93312-93375
START_RANGE client47a socket1495 620 93375-93437
START_RANGE client47a socket1496 630 93437-93500
START_RANGE client47a socket1497 620 93500-93562
START_RANGE client47a socket1498 630 93562-93625
START_RANGE client47a socket1499 620 93625-93687
START_RANGE client47a socket1500 630 93687-93750
START_RANGE client47a socket1501 620 93750-93812
START_RANGE client47a socket1502 630 93812-93875
START_RANGE client47a socket1503 620 93875-93937
START_RANGE client47a socket1504 630 93937-94000
START_RANGE client48a socket1505 620 94000-94062
START_RANGE client48a socket1506 630 94062-94125
START_RANGE client48a socket1507 620 94125-94187
START_RANGE client48a socket1508 630 94187-94250
START_RANGE client48a socket1509 620 94250-94312
START_RANGE client48a socket1510 630 94312-94375
START_RANGE client48a socket1511 620 94375-94437
START_RANGE client48a socket1512 630 94437-94500
START_RANGE client48a socket1513 620 94500-94562
START_RANGE client48a socket1514 630 94562-94625
START_RANGE client48a socket1515 620 94625-94687
START_RANGE client48a socket1516 630 94687-94750
START_RANGE client48a socket1517 620 94750-94812
START_RANGE client48a socket1518 630 94812-94875
START_RANGE client48a socket1519 620 94875-94937
START_RANGE client48a socket1520 630 94937-95000
START_RANGE client48a socket1521 620 95000-95062
START_RANGE client48a socket1522 630 95062-95125
START_RANGE client48a socket1523 620 95125-95187
START_RANGE client48a socket1524 630 95187-95250
START_RANGE client48a socket1525 620 95250-95312
START_RANGE client48a socket1526 630 95312-95375
START_RANGE client48a socket1527 620 95375-95437
START_RANGE client48a socket1528 630 95437-95500
START_RANGE client48a socket1529 620 95500-95562
START_RANGE client48a socket1530 630 95562-95625
START_RANGE client48a socket1531 620 95625-95687
START_RANGE client48a socket1532 630 95687-95750
START_RANGE client48a socket1533 620 95750-95812
START_RANGE client48a socket1534 630 95812-95875
START_RANGE client48a socket1535 620 95875-95937
START_RANGE client48a socket1536 630 95937-96000
#elif MASTER_NUM13
START_RANGE client49a socket1537 620 96000-96062
START_RANGE client49a socket1538 630 96062-96125
START_RANGE client49a socket1539 620 96125-96187
START_RANGE client49a socket1540 630 96187-96250
START_RANGE client49a socket1541 620 96250-96312
START_RANGE client49a socket1542 630 96312-96375
START_RANGE client49a socket1543 620 96375-96437
START_RANGE client49a socket1544 630 96437-96500
START_RANGE client49a socket1545 620 96500-96562
START_RANGE client49a socket1546 630 96562-96625
START_RANGE client49a socket1547 620 96625-96687

START_RANGE client49a socket1548 630 96687-96750
START_RANGE client49a socket1549 620 96750-96812
START_RANGE client49a socket1550 630 96812-96875
START_RANGE client49a socket1551 620 96875-96937
START_RANGE client49a socket1552 630 96937-97000
START_RANGE client49a socket1553 620 97000-97062
START_RANGE client49a socket1554 630 97062-97125
START_RANGE client49a socket1555 620 97125-97187
START_RANGE client49a socket1556 630 97187-97250
START_RANGE client49a socket1557 620 97250-97312
START_RANGE client49a socket1558 630 97312-97375
START_RANGE client49a socket1559 620 97375-97437
START_RANGE client49a socket1560 630 97437-97500
START_RANGE client49a socket1561 620 97500-97562
START_RANGE client49a socket1562 630 97562-97625
START_RANGE client49a socket1563 620 97625-97687
START_RANGE client49a socket1564 630 97687-97750
START_RANGE client49a socket1565 620 97750-97812
START_RANGE client49a socket1566 630 97812-97875
START_RANGE client49a socket1567 620 97875-97937
START_RANGE client49a socket1568 630 97937-98000
START_RANGE client50a socket1569 620 98000-98062
START_RANGE client50a socket1570 630 98062-98125
START_RANGE client50a socket1571 620 98125-98187
START_RANGE client50a socket1572 630 98187-98250
START_RANGE client50a socket1573 620 98250-98312
START_RANGE client50a socket1574 630 98312-98375
START_RANGE client50a socket1575 620 98375-98437
START_RANGE client50a socket1576 630 98437-98500
START_RANGE client50a socket1577 620 98500-98562
START_RANGE client50a socket1578 630 98562-98625
START_RANGE client50a socket1579 620 98625-98687
START_RANGE client50a socket1580 630 98687-98750
START_RANGE client50a socket1581 620 98750-98812
START_RANGE client50a socket1582 630 98812-98875
START_RANGE client50a socket1583 620 98875-98937
START_RANGE client50a socket1584 630 98937-99000
START_RANGE client50a socket1585 620 99000-99062
START_RANGE client50a socket1586 630 99062-99125
START_RANGE client50a socket1587 620 99125-99187
START_RANGE client50a socket1588 630 99187-99250
START_RANGE client50a socket1589 620 99250-99312
START_RANGE client50a socket1590 630 99312-99375
START_RANGE client50a socket1591 620 99375-99437
START_RANGE client50a socket1592 630 99437-99500
START_RANGE client50a socket1593 620 99500-99562
START_RANGE client50a socket1594 630 99562-99625
START_RANGE client50a socket1595 620 99625-99687
START_RANGE client50a socket1596 630 99687-99750
START_RANGE client50a socket1597 620 99750-99812
START_RANGE client50a socket1598 630 99812-99875
START_RANGE client50a socket1599 620 99875-99937
START_RANGE client50a socket1600 630 99937-100000
START_RANGE client51a socket1601 620 100000-100062
START_RANGE client51a socket1602 630 100062-100125
START_RANGE client51a socket1603 620 100125-100187
START_RANGE client51a socket1604 630 100187-100250
START_RANGE client51a socket1605 620 100250-100312
START_RANGE client51a socket1606 630 100312-100375
START_RANGE client51a socket1607 620 100375-100437
START_RANGE client51a socket1608 630 100437-100500
START_RANGE client51a socket1609 620 100500-100562

START_RANGE client51a socket1610 630 100562-100625
START_RANGE client51a socket1611 620 100625-100687
START_RANGE client51a socket1612 630 100687-100750
START_RANGE client51a socket1613 620 100750-100812
START_RANGE client51a socket1614 630 100812-100875
START_RANGE client51a socket1615 620 100875-100937
START_RANGE client51a socket1616 630 100937-101000
START_RANGE client51a socket1617 620 101000-101062
START_RANGE client51a socket1618 630 101062-101125
START_RANGE client51a socket1619 620 101125-101187
START_RANGE client51a socket1620 630 101187-101250
START_RANGE client51a socket1621 620 101250-101312
START_RANGE client51a socket1622 630 101312-101375
START_RANGE client51a socket1623 620 101375-101437
START_RANGE client51a socket1624 630 101437-101500
START_RANGE client51a socket1625 620 101500-101562
START_RANGE client51a socket1626 630 101562-101625
START_RANGE client51a socket1627 620 101625-101687
START_RANGE client51a socket1628 630 101687-101750
START_RANGE client51a socket1629 620 101750-101812
START_RANGE client51a socket1630 630 101812-101875
START_RANGE client51a socket1631 620 101875-101937
START_RANGE client51a socket1632 630 101937-102000
START_RANGE client52a socket1633 620 102000-102062
START_RANGE client52a socket1634 630 102062-102125
START_RANGE client52a socket1635 620 102125-102187
START_RANGE client52a socket1636 630 102187-102250
START_RANGE client52a socket1637 620 102250-102312
START_RANGE client52a socket1638 630 102312-102375
START_RANGE client52a socket1639 620 102375-102437
START_RANGE client52a socket1640 630 102437-102500
START_RANGE client52a socket1641 620 102500-102562
START_RANGE client52a socket1642 630 102562-102625
START_RANGE client52a socket1643 620 102625-102687
START_RANGE client52a socket1644 630 102687-102750
START_RANGE client52a socket1645 620 102750-102812
START_RANGE client52a socket1646 630 102812-102875
START_RANGE client52a socket1647 620 102875-102937
START_RANGE client52a socket1648 630 102937-103000
START_RANGE client52a socket1649 620 103000-103062
START_RANGE client52a socket1650 630 103062-103125
START_RANGE client52a socket1651 620 103125-103187
START_RANGE client52a socket1652 630 103187-103250
START_RANGE client52a socket1653 620 103250-103312
START_RANGE client52a socket1654 630 103312-103375
START_RANGE client52a socket1655 620 103375-103437
START_RANGE client52a socket1656 630 103437-103500
START_RANGE client52a socket1657 620 103500-103562
START_RANGE client52a socket1658 630 103562-103625
START_RANGE client52a socket1659 620 103625-103687
START_RANGE client52a socket1660 630 103687-103750
START_RANGE client52a socket1661 620 103750-103812
START_RANGE client52a socket1662 630 103812-103875
START_RANGE client52a socket1663 620 103875-103937
START_RANGE client52a socket1664 630 103937-104000
#elif MASTER_NUM14
START_RANGE client53a socket1665 620 104000-104062
START_RANGE client53a socket1666 630 104062-104125
START_RANGE client53a socket1667 620 104125-104187
START_RANGE client53a socket1668 630 104187-104250
START_RANGE client53a socket1669 620 104250-104312
START_RANGE client53a socket1670 630 104312-104375

START_RANGE client53a socket1671 620 104375-104437
START_RANGE client53a socket1672 630 104437-104500
START_RANGE client53a socket1673 620 104500-104562
START_RANGE client53a socket1674 630 104562-104625
START_RANGE client53a socket1675 620 104625-104687
START_RANGE client53a socket1676 630 104687-104750
START_RANGE client53a socket1677 620 104750-104812
START_RANGE client53a socket1678 630 104812-104875
START_RANGE client53a socket1679 620 104875-104937
START_RANGE client53a socket1680 630 104937-105000
START_RANGE client53a socket1681 620 105000-105062
START_RANGE client53a socket1682 630 105062-105125
START_RANGE client53a socket1683 620 105125-105187
START_RANGE client53a socket1684 630 105187-105250
START_RANGE client53a socket1685 620 105250-105312
START_RANGE client53a socket1686 630 105312-105375
START_RANGE client53a socket1687 620 105375-105437
START_RANGE client53a socket1688 630 105437-105500
START_RANGE client53a socket1689 620 105500-105562
START_RANGE client53a socket1690 630 105562-105625
START_RANGE client53a socket1691 620 105625-105687
START_RANGE client53a socket1692 630 105687-105750
START_RANGE client53a socket1693 620 105750-105812
START_RANGE client53a socket1694 630 105812-105875
START_RANGE client53a socket1695 620 105875-105937
START_RANGE client53a socket1696 630 105937-106000
START_RANGE client54a socket1697 620 106000-106062
START_RANGE client54a socket1698 630 106062-106125
START_RANGE client54a socket1699 620 106125-106187
START_RANGE client54a socket1700 630 106187-106250
START_RANGE client54a socket1701 620 106250-106312
START_RANGE client54a socket1702 630 106312-106375
START_RANGE client54a socket1703 620 106375-106437
START_RANGE client54a socket1704 630 106437-106500
START_RANGE client54a socket1705 620 106500-106562
START_RANGE client54a socket1706 630 106562-106625
START_RANGE client54a socket1707 620 106625-106687
START_RANGE client54a socket1708 630 106687-106750
START_RANGE client54a socket1709 620 106750-106812
START_RANGE client54a socket1710 630 106812-106875
START_RANGE client54a socket1711 620 106875-106937
START_RANGE client54a socket1712 630 106937-107000
START_RANGE client54a socket1713 620 107000-107062
START_RANGE client54a socket1714 630 107062-107125
START_RANGE client54a socket1715 620 107125-107187
START_RANGE client54a socket1716 630 107187-107250
START_RANGE client54a socket1717 620 107250-107312
START_RANGE client54a socket1718 630 107312-107375
START_RANGE client54a socket1719 620 107375-107437
START_RANGE client54a socket1720 630 107437-107500
START_RANGE client54a socket1721 620 107500-107562
START_RANGE client54a socket1722 630 107562-107625
START_RANGE client54a socket1723 620 107625-107687
START_RANGE client54a socket1724 630 107687-107750
START_RANGE client54a socket1725 620 107750-107812
START_RANGE client54a socket1726 630 107812-107875
START_RANGE client54a socket1727 620 107875-107937
START_RANGE client54a socket1728 630 107937-108000
START_RANGE client55a socket1729 620 108000-108062
START_RANGE client55a socket1730 630 108062-108125
START_RANGE client55a socket1731 620 108125-108187
START_RANGE client55a socket1732 630 108187-108250

START_RANGE client55a socket1733 620 108250-108312
START_RANGE client55a socket1734 630 108312-108375
START_RANGE client55a socket1735 620 108375-108437
START_RANGE client55a socket1736 630 108437-108500
START_RANGE client55a socket1737 620 108500-108562
START_RANGE client55a socket1738 630 108562-108625
START_RANGE client55a socket1739 620 108625-108687
START_RANGE client55a socket1740 630 108687-108750
START_RANGE client55a socket1741 620 108750-108812
START_RANGE client55a socket1742 630 108812-108875
START_RANGE client55a socket1743 620 108875-108937
START_RANGE client55a socket1744 630 108937-109000
START_RANGE client55a socket1745 620 109000-109062
START_RANGE client55a socket1746 630 109062-109125
START_RANGE client55a socket1747 620 109125-109187
START_RANGE client55a socket1748 630 109187-109250
START_RANGE client55a socket1749 620 109250-109312
START_RANGE client55a socket1750 630 109312-109375
START_RANGE client55a socket1751 620 109375-109437
START_RANGE client55a socket1752 630 109437-109500
START_RANGE client55a socket1753 620 109500-109562
START_RANGE client55a socket1754 630 109562-109625
START_RANGE client55a socket1755 620 109625-109687
START_RANGE client55a socket1756 630 109687-109750
START_RANGE client55a socket1757 620 109750-109812
START_RANGE client55a socket1758 630 109812-109875
START_RANGE client55a socket1759 620 109875-109937
START_RANGE client55a socket1760 630 109937-110000
START_RANGE client55a socket1761 620 110000-110062
START_RANGE client55a socket1762 630 110062-110125
START_RANGE client55a socket1763 620 110125-110187
START_RANGE client55a socket1764 630 110187-110250
START_RANGE client55a socket1765 620 110250-110312
START_RANGE client55a socket1766 630 110312-110375
START_RANGE client55a socket1767 620 110375-110437
START_RANGE client55a socket1768 630 110437-110500
START_RANGE client55a socket1769 620 110500-110562
START_RANGE client55a socket1770 630 110562-110625
START_RANGE client55a socket1771 620 110625-110687
START_RANGE client55a socket1772 630 110687-110750
START_RANGE client55a socket1773 620 110750-110812
START_RANGE client55a socket1774 630 110812-110875
START_RANGE client55a socket1775 620 110875-110937
START_RANGE client55a socket1776 630 110937-111000
START_RANGE client55a socket1777 620 111000-111062
START_RANGE client55a socket1778 630 111062-111125
START_RANGE client55a socket1779 620 111125-111187
START_RANGE client55a socket1780 630 111187-111250
START_RANGE client55a socket1781 620 111250-111312
START_RANGE client55a socket1782 630 111312-111375
START_RANGE client55a socket1783 620 111375-111437
START_RANGE client55a socket1784 630 111437-111500
START_RANGE client55a socket1785 620 111500-111562
START_RANGE client55a socket1786 630 111562-111625
START_RANGE client55a socket1787 620 111625-111687
START_RANGE client55a socket1788 630 111687-111750
START_RANGE client55a socket1789 620 111750-111812
START_RANGE client55a socket1790 630 111812-111875
START_RANGE client55a socket1791 620 111875-111937
START_RANGE client55a socket1792 630 111937-112000
#elif MASTER_NUM15
START_RANGE client57a socket1793 620 112000-112062

START_RANGE client57a socket1794 630 112062-112125
START_RANGE client57a socket1795 620 112125-112187
START_RANGE client57a socket1796 630 112187-112250
START_RANGE client57a socket1797 620 112250-112312
START_RANGE client57a socket1798 630 112312-112375
START_RANGE client57a socket1799 620 112375-112437
START_RANGE client57a socket1800 630 112437-112500
START_RANGE client57a socket1801 620 112500-112562
START_RANGE client57a socket1802 630 112562-112625
START_RANGE client57a socket1803 620 112625-112687
START_RANGE client57a socket1804 630 112687-112750
START_RANGE client57a socket1805 620 112750-112812
START_RANGE client57a socket1806 630 112812-112875
START_RANGE client57a socket1807 620 112875-112937
START_RANGE client57a socket1808 630 112937-113000
START_RANGE client57a socket1809 620 113000-113062
START_RANGE client57a socket1810 630 113062-113125
START_RANGE client57a socket1811 620 113125-113187
START_RANGE client57a socket1812 630 113187-113250
START_RANGE client57a socket1813 620 113250-113312
START_RANGE client57a socket1814 630 113312-113375
START_RANGE client57a socket1815 620 113375-113437
START_RANGE client57a socket1816 630 113437-113500
START_RANGE client57a socket1817 620 113500-113562
START_RANGE client57a socket1818 630 113562-113625
START_RANGE client57a socket1819 620 113625-113687
START_RANGE client57a socket1820 630 113687-113750
START_RANGE client57a socket1821 620 113750-113812
START_RANGE client57a socket1822 630 113812-113875
START_RANGE client57a socket1823 620 113875-113937
START_RANGE client57a socket1824 630 113937-114000
START_RANGE client58a socket1825 620 114000-114062
START_RANGE client58a socket1826 630 114062-114125
START_RANGE client58a socket1827 620 114125-114187
START_RANGE client58a socket1828 630 114187-114250
START_RANGE client58a socket1829 620 114250-114312
START_RANGE client58a socket1830 630 114312-114375
START_RANGE client58a socket1831 620 114375-114437
START_RANGE client58a socket1832 630 114437-114500
START_RANGE client58a socket1833 620 114500-114562
START_RANGE client58a socket1834 630 114562-114625
START_RANGE client58a socket1835 620 114625-114687
START_RANGE client58a socket1836 630 114687-114750
START_RANGE client58a socket1837 620 114750-114812
START_RANGE client58a socket1838 630 114812-114875
START_RANGE client58a socket1839 620 114875-114937
START_RANGE client58a socket1840 630 114937-115000
START_RANGE client58a socket1841 620 115000-115062
START_RANGE client58a socket1842 630 115062-115125
START_RANGE client58a socket1843 620 115125-115187
START_RANGE client58a socket1844 630 115187-115250
START_RANGE client58a socket1845 620 115250-115312
START_RANGE client58a socket1846 630 115312-115375
START_RANGE client58a socket1847 620 115375-115437
START_RANGE client58a socket1848 630 115437-115500
START_RANGE client58a socket1849 620 115500-115562
START_RANGE client58a socket1850 630 115562-115625
START_RANGE client58a socket1851 620 115625-115687
START_RANGE client58a socket1852 630 115687-115750
START_RANGE client58a socket1853 620 115750-115812
START_RANGE client58a socket1854 630 115812-115875
START_RANGE client58a socket1855 620 115875-115937

START_RANGE client58a socket1856 630 115937-116000
START_RANGE client59a socket1857 620 116000-116062
START_RANGE client59a socket1858 630 116062-116125
START_RANGE client59a socket1859 620 116125-116187
START_RANGE client59a socket1860 630 116187-116250
START_RANGE client59a socket1861 620 116250-116312
START_RANGE client59a socket1862 630 116312-116375
START_RANGE client59a socket1863 620 116375-116437
START_RANGE client59a socket1864 630 116437-116500
START_RANGE client59a socket1865 620 116500-116562
START_RANGE client59a socket1866 630 116562-116625
START_RANGE client59a socket1867 620 116625-116687
START_RANGE client59a socket1868 630 116687-116750
START_RANGE client59a socket1869 620 116750-116812
START_RANGE client59a socket1870 630 116812-116875
START_RANGE client59a socket1871 620 116875-116937
START_RANGE client59a socket1872 630 116937-117000
START_RANGE client59a socket1873 620 117000-117062
START_RANGE client59a socket1874 630 117062-117125
START_RANGE client59a socket1875 620 117125-117187
START_RANGE client59a socket1876 630 117187-117250
START_RANGE client59a socket1877 620 117250-117312
START_RANGE client59a socket1878 630 117312-117375
START_RANGE client59a socket1879 620 117375-117437
START_RANGE client59a socket1880 630 117437-117500
START_RANGE client59a socket1881 620 117500-117562
START_RANGE client59a socket1882 630 117562-117625
START_RANGE client59a socket1883 620 117625-117687
START_RANGE client59a socket1884 630 117687-117750
START_RANGE client59a socket1885 620 117750-117812
START_RANGE client59a socket1886 630 117812-117875
START_RANGE client59a socket1887 620 117875-117937
START_RANGE client59a socket1888 630 117937-118000
START_RANGE client60a socket1889 620 118000-118062
START_RANGE client60a socket1890 630 118062-118125
START_RANGE client60a socket1891 620 118125-118187
START_RANGE client60a socket1892 630 118187-118250
START_RANGE client60a socket1893 620 118250-118312
START_RANGE client60a socket1894 630 118312-118375
START_RANGE client60a socket1895 620 118375-118437
START_RANGE client60a socket1896 630 118437-118500
START_RANGE client60a socket1897 620 118500-118562
START_RANGE client60a socket1898 630 118562-118625
START_RANGE client60a socket1899 620 118625-118687
START_RANGE client60a socket1900 630 118687-118750
START_RANGE client60a socket1901 620 118750-118812
START_RANGE client60a socket1902 630 118812-118875
START_RANGE client60a socket1903 620 118875-118937
START_RANGE client60a socket1904 630 118937-119000
START_RANGE client60a socket1905 620 119000-119062
START_RANGE client60a socket1906 630 119062-119125
START_RANGE client60a socket1907 620 119125-119187
START_RANGE client60a socket1908 630 119187-119250
START_RANGE client60a socket1909 620 119250-119312
START_RANGE client60a socket1910 630 119312-119375
START_RANGE client60a socket1911 620 119375-119437
START_RANGE client60a socket1912 630 119437-119500
START_RANGE client60a socket1913 620 119500-119562
START_RANGE client60a socket1914 630 119562-119625
START_RANGE client60a socket1915 620 119625-119687
START_RANGE client60a socket1916 630 119687-119750
START_RANGE client60a socket1917 620 119750-119812

START_RANGE client60a socket1918 630 119812-119875
START_RANGE client60a socket1919 620 119875-119937
START_RANGE client60a socket1920 630 119937-120000
#elif MASTER_NUM16
START_RANGE client61a socket1921 620 120000-120062
START_RANGE client61a socket1922 630 120062-120125
START_RANGE client61a socket1923 620 120125-120187
START_RANGE client61a socket1924 630 120187-120250
START_RANGE client61a socket1925 620 120250-120312
START_RANGE client61a socket1926 630 120312-120375
START_RANGE client61a socket1927 620 120375-120437
START_RANGE client61a socket1928 630 120437-120500
START_RANGE client61a socket1929 620 120500-120562
START_RANGE client61a socket1930 630 120562-120625
START_RANGE client61a socket1931 620 120625-120687
START_RANGE client61a socket1932 630 120687-120750
START_RANGE client61a socket1933 620 120750-120812
START_RANGE client61a socket1934 630 120812-120875
START_RANGE client61a socket1935 620 120875-120937
START_RANGE client61a socket1936 630 120937-121000
START_RANGE client61a socket1937 620 121000-121062
START_RANGE client61a socket1938 630 121062-121125
START_RANGE client61a socket1939 620 121125-121187
START_RANGE client61a socket1940 630 121187-121250
START_RANGE client61a socket1941 620 121250-121312
START_RANGE client61a socket1942 630 121312-121375
START_RANGE client61a socket1943 620 121375-121437
START_RANGE client61a socket1944 630 121437-121500
START_RANGE client61a socket1945 620 121500-121562
START_RANGE client61a socket1946 630 121562-121625
START_RANGE client61a socket1947 620 121625-121687
START_RANGE client61a socket1948 630 121687-121750
START_RANGE client61a socket1949 620 121750-121812
START_RANGE client61a socket1950 630 121812-121875
START_RANGE client61a socket1951 620 121875-121937
START_RANGE client61a socket1952 630 121937-122000
START_RANGE client62a socket1953 620 122000-122062
START_RANGE client62a socket1954 630 122062-122125
START_RANGE client62a socket1955 620 122125-122187
START_RANGE client62a socket1956 630 122187-122250
START_RANGE client62a socket1957 620 122250-122312
START_RANGE client62a socket1958 630 122312-122375
START_RANGE client62a socket1959 620 122375-122437
START_RANGE client62a socket1960 630 122437-122500
START_RANGE client62a socket1961 620 122500-122562
START_RANGE client62a socket1962 630 122562-122625
START_RANGE client62a socket1963 620 122625-122687
START_RANGE client62a socket1964 630 122687-122750
START_RANGE client62a socket1965 620 122750-122812
START_RANGE client62a socket1966 630 122812-122875
START_RANGE client62a socket1967 620 122875-122937
START_RANGE client62a socket1968 630 122937-123000
START_RANGE client62a socket1969 620 123000-123062
START_RANGE client62a socket1970 630 123062-123125
START_RANGE client62a socket1971 620 123125-123187
START_RANGE client62a socket1972 630 123187-123250
START_RANGE client62a socket1973 620 123250-123312
START_RANGE client62a socket1974 630 123312-123375
START_RANGE client62a socket1975 620 123375-123437
START_RANGE client62a socket1976 630 123437-123500
START_RANGE client62a socket1977 620 123500-123562
START_RANGE client62a socket1978 630 123562-123625

START_RANGE client62a socket1979 620 123625-123687
START_RANGE client62a socket1980 630 123687-123750
START_RANGE client62a socket1981 620 123750-123812
START_RANGE client62a socket1982 630 123812-123875
START_RANGE client62a socket1983 620 123875-123937
START_RANGE client62a socket1984 630 123937-124000
START_RANGE client63a socket1985 620 124000-124062
START_RANGE client63a socket1986 630 124062-124125
START_RANGE client63a socket1987 620 124125-124187
START_RANGE client63a socket1988 630 124187-124250
START_RANGE client63a socket1989 620 124250-124312
START_RANGE client63a socket1990 630 124312-124375
START_RANGE client63a socket1991 620 124375-124437
START_RANGE client63a socket1992 630 124437-124500
START_RANGE client63a socket1993 620 124500-124562
START_RANGE client63a socket1994 630 124562-124625
START_RANGE client63a socket1995 620 124625-124687
START_RANGE client63a socket1996 630 124687-124750
START_RANGE client63a socket1997 620 124750-124812
START_RANGE client63a socket1998 630 124812-124875
START_RANGE client63a socket1999 620 124875-124937
START_RANGE client63a socket2000 630 124937-125000
START_RANGE client63a socket2001 620 125000-125062
START_RANGE client63a socket2002 630 125062-125125
START_RANGE client63a socket2003 620 125125-125187
START_RANGE client63a socket2004 630 125187-125250
START_RANGE client63a socket2005 620 125250-125312
START_RANGE client63a socket2006 630 125312-125375
START_RANGE client63a socket2007 620 125375-125437
START_RANGE client63a socket2008 630 125437-125500
START_RANGE client63a socket2009 620 125500-125562
START_RANGE client63a socket2010 630 125562-125625
START_RANGE client63a socket2011 620 125625-125687
START_RANGE client63a socket2012 630 125687-125750
START_RANGE client63a socket2013 620 125750-125812
START_RANGE client63a socket2014 630 125812-125875
START_RANGE client63a socket2015 620 125875-125937
START_RANGE client63a socket2016 630 125937-126000
START_RANGE client64a socket2017 620 126000-126062
START_RANGE client64a socket2018 630 126062-126125
START_RANGE client64a socket2019 620 126125-126187
START_RANGE client64a socket2020 630 126187-126250
START_RANGE client64a socket2021 620 126250-126312
START_RANGE client64a socket2022 630 126312-126375
START_RANGE client64a socket2023 620 126375-126437
START_RANGE client64a socket2024 630 126437-126500
START_RANGE client64a socket2025 620 126500-126562
START_RANGE client64a socket2026 630 126562-126625
START_RANGE client64a socket2027 620 126625-126687
START_RANGE client64a socket2028 630 126687-126750
START_RANGE client64a socket2029 620 126750-126812
START_RANGE client64a socket2030 630 126812-126875
START_RANGE client64a socket2031 620 126875-126937
START_RANGE client64a socket2032 630 126937-127000
START_RANGE client64a socket2033 620 127000-127062
START_RANGE client64a socket2034 630 127062-127125
START_RANGE client64a socket2035 620 127125-127187
START_RANGE client64a socket2036 630 127187-127250
START_RANGE client64a socket2037 620 127250-127312
START_RANGE client64a socket2038 630 127312-127375
START_RANGE client64a socket2039 620 127375-127437
START_RANGE client64a socket2040 630 127437-127500

START_RANGE client64a socket2041 620 127500-127562
START_RANGE client64a socket2042 630 127562-127625
START_RANGE client64a socket2043 620 127625-127687
START_RANGE client64a socket2044 630 127687-127750
START_RANGE client64a socket2045 620 127750-127812
START_RANGE client64a socket2046 630 127812-127875
START_RANGE client64a socket2047 620 127875-127937
START_RANGE client64a socket2048 630 127937-128000
#elif MASTER_NUM17
START_RANGE client65a socket2049 620 128000-128062
START_RANGE client65a socket2050 630 128062-128125
START_RANGE client65a socket2051 620 128125-128187
START_RANGE client65a socket2052 630 128187-128250
START_RANGE client65a socket2053 620 128250-128312
START_RANGE client65a socket2054 630 128312-128375
START_RANGE client65a socket2055 620 128375-128437
START_RANGE client65a socket2056 630 128437-128500
START_RANGE client65a socket2057 620 128500-128562
START_RANGE client65a socket2058 630 128562-128625
START_RANGE client65a socket2059 620 128625-128687
START_RANGE client65a socket2060 630 128687-128750
START_RANGE client65a socket2061 620 128750-128812
START_RANGE client65a socket2062 630 128812-128875
START_RANGE client65a socket2063 620 128875-128937
START_RANGE client65a socket2064 630 128937-129000
START_RANGE client65a socket2065 620 129000-129062
START_RANGE client65a socket2066 630 129062-129125
START_RANGE client65a socket2067 620 129125-129187
START_RANGE client65a socket2068 630 129187-129250
START_RANGE client65a socket2069 620 129250-129312
START_RANGE client65a socket2070 630 129312-129375
START_RANGE client65a socket2071 620 129375-129437
START_RANGE client65a socket2072 630 129437-129500
START_RANGE client65a socket2073 620 129500-129562
START_RANGE client65a socket2074 630 129562-129625
START_RANGE client65a socket2075 620 129625-129687
START_RANGE client65a socket2076 630 129687-129750
START_RANGE client65a socket2077 620 129750-129812
START_RANGE client65a socket2078 630 129812-129875
START_RANGE client65a socket2079 620 129875-129937
START_RANGE client65a socket2080 630 129937-130000
START_RANGE client66a socket2081 620 130000-130062
START_RANGE client66a socket2082 630 130062-130125
START_RANGE client66a socket2083 620 130125-130187
START_RANGE client66a socket2084 630 130187-130250
START_RANGE client66a socket2085 620 130250-130312
START_RANGE client66a socket2086 630 130312-130375
START_RANGE client66a socket2087 620 130375-130437
START_RANGE client66a socket2088 630 130437-130500
START_RANGE client66a socket2089 620 130500-130562
START_RANGE client66a socket2090 630 130562-130625
START_RANGE client66a socket2091 620 130625-130687
START_RANGE client66a socket2092 630 130687-130750
START_RANGE client66a socket2093 620 130750-130812
START_RANGE client66a socket2094 630 130812-130875
START_RANGE client66a socket2095 620 130875-130937
START_RANGE client66a socket2096 630 130937-131000
START_RANGE client66a socket2097 620 131000-131062
START_RANGE client66a socket2098 630 131062-131125
START_RANGE client66a socket2099 620 131125-131187
START_RANGE client66a socket2100 630 131187-131250
START_RANGE client66a socket2101 620 131250-131312

START_RANGE client66a socket2102 630 131312-131375
START_RANGE client66a socket2103 620 131375-131437
START_RANGE client66a socket2104 630 131437-131500
START_RANGE client66a socket2105 620 131500-131562
START_RANGE client66a socket2106 630 131562-131625
START_RANGE client66a socket2107 620 131625-131687
START_RANGE client66a socket2108 630 131687-131750
START_RANGE client66a socket2109 620 131750-131812
START_RANGE client66a socket2110 630 131812-131875
START_RANGE client66a socket2111 620 131875-131937
START_RANGE client66a socket2112 630 131937-132000
START_RANGE client67a socket2113 620 132000-132062
START_RANGE client67a socket2114 630 132062-132125
START_RANGE client67a socket2115 620 132125-132187
START_RANGE client67a socket2116 630 132187-132250
START_RANGE client67a socket2117 620 132250-132312
START_RANGE client67a socket2118 630 132312-132375
START_RANGE client67a socket2119 620 132375-132437
START_RANGE client67a socket2120 630 132437-132500
START_RANGE client67a socket2121 620 132500-132562
START_RANGE client67a socket2122 630 132562-132625
START_RANGE client67a socket2123 620 132625-132687
START_RANGE client67a socket2124 630 132687-132750
START_RANGE client67a socket2125 620 132750-132812
START_RANGE client67a socket2126 630 132812-132875
START_RANGE client67a socket2127 620 132875-132937
START_RANGE client67a socket2128 630 132937-133000
START_RANGE client67a socket2129 620 133000-133062
START_RANGE client67a socket2130 630 133062-133125
START_RANGE client67a socket2131 620 133125-133187
START_RANGE client67a socket2132 630 133187-133250
START_RANGE client67a socket2133 620 133250-133312
START_RANGE client67a socket2134 630 133312-133375
START_RANGE client67a socket2135 620 133375-133437
START_RANGE client67a socket2136 630 133437-133500
START_RANGE client67a socket2137 620 133500-133562
START_RANGE client67a socket2138 630 133562-133625
START_RANGE client67a socket2139 620 133625-133687
START_RANGE client67a socket2140 630 133687-133750
START_RANGE client67a socket2141 620 133750-133812
START_RANGE client67a socket2142 630 133812-133875
START_RANGE client67a socket2143 620 133875-133937
START_RANGE client67a socket2144 630 133937-134000
START_RANGE client68a socket2145 620 134000-134062
START_RANGE client68a socket2146 630 134062-134125
START_RANGE client68a socket2147 620 134125-134187
START_RANGE client68a socket2148 630 134187-134250
START_RANGE client68a socket2149 620 134250-134312
START_RANGE client68a socket2150 630 134312-134375
START_RANGE client68a socket2151 620 134375-134437
START_RANGE client68a socket2152 630 134437-134500
START_RANGE client68a socket2153 620 134500-134562
START_RANGE client68a socket2154 630 134562-134625
START_RANGE client68a socket2155 620 134625-134687
START_RANGE client68a socket2156 630 134687-134750
START_RANGE client68a socket2157 620 134750-134812
START_RANGE client68a socket2158 630 134812-134875
START_RANGE client68a socket2159 620 134875-134937
START_RANGE client68a socket2160 630 134937-135000
START_RANGE client68a socket2161 620 135000-135062
START_RANGE client68a socket2162 630 135062-135125
START_RANGE client68a socket2163 620 135125-135187

START_RANGE client68a socket2164 630 135187-135250
START_RANGE client68a socket2165 620 135250-135312
START_RANGE client68a socket2166 630 135312-135375
START_RANGE client68a socket2167 620 135375-135437
START_RANGE client68a socket2168 630 135437-135500
START_RANGE client68a socket2169 620 135500-135562
START_RANGE client68a socket2170 630 135562-135625
START_RANGE client68a socket2171 620 135625-135687
START_RANGE client68a socket2172 630 135687-135750
START_RANGE client68a socket2173 620 135750-135812
START_RANGE client68a socket2174 630 135812-135875
START_RANGE client68a socket2175 620 135875-135937
START_RANGE client68a socket2176 630 135937-136000
#elif MASTER_NUM18
START_RANGE client69a socket2177 620 136000-136062
START_RANGE client69a socket2178 630 136062-136125
START_RANGE client69a socket2179 620 136125-136187
START_RANGE client69a socket2180 630 136187-136250
START_RANGE client69a socket2181 620 136250-136312
START_RANGE client69a socket2182 630 136312-136375
START_RANGE client69a socket2183 620 136375-136437
START_RANGE client69a socket2184 630 136437-136500
START_RANGE client69a socket2185 620 136500-136562
START_RANGE client69a socket2186 630 136562-136625
START_RANGE client69a socket2187 620 136625-136687
START_RANGE client69a socket2188 630 136687-136750
START_RANGE client69a socket2189 620 136750-136812
START_RANGE client69a socket2190 630 136812-136875
START_RANGE client69a socket2191 620 136875-136937
START_RANGE client69a socket2192 630 136937-137000
START_RANGE client69a socket2193 620 137000-137062
START_RANGE client69a socket2194 630 137062-137125
START_RANGE client69a socket2195 620 137125-137187
START_RANGE client69a socket2196 630 137187-137250
START_RANGE client69a socket2197 620 137250-137312
START_RANGE client69a socket2198 630 137312-137375
START_RANGE client69a socket2199 620 137375-137437
START_RANGE client69a socket2200 630 137437-137500
START_RANGE client69a socket2201 620 137500-137562
START_RANGE client69a socket2202 630 137562-137625
START_RANGE client69a socket2203 620 137625-137687
START_RANGE client69a socket2204 630 137687-137750
START_RANGE client69a socket2205 620 137750-137812
START_RANGE client69a socket2206 630 137812-137875
START_RANGE client69a socket2207 620 137875-137937
START_RANGE client69a socket2208 630 137937-138000
START_RANGE client70a socket2209 620 138000-138062
START_RANGE client70a socket2210 630 138062-138125
START_RANGE client70a socket2211 620 138125-138187
START_RANGE client70a socket2212 630 138187-138250
START_RANGE client70a socket2213 620 138250-138312
START_RANGE client70a socket2214 630 138312-138375
START_RANGE client70a socket2215 620 138375-138437
START_RANGE client70a socket2216 630 138437-138500
START_RANGE client70a socket2217 620 138500-138562
START_RANGE client70a socket2218 630 138562-138625
START_RANGE client70a socket2219 620 138625-138687
START_RANGE client70a socket2220 630 138687-138750
START_RANGE client70a socket2221 620 138750-138812
START_RANGE client70a socket2222 630 138812-138875
START_RANGE client70a socket2223 620 138875-138937
START_RANGE client70a socket2224 630 138937-139000

START_RANGE client70a socket2225 620 139000-139062
START_RANGE client70a socket2226 630 139062-139125
START_RANGE client70a socket2227 620 139125-139187
START_RANGE client70a socket2228 630 139187-139250
START_RANGE client70a socket2229 620 139250-139312
START_RANGE client70a socket2230 630 139312-139375
START_RANGE client70a socket2231 620 139375-139437
START_RANGE client70a socket2232 630 139437-139500
START_RANGE client70a socket2233 620 139500-139562
START_RANGE client70a socket2234 630 139562-139625
START_RANGE client70a socket2235 620 139625-139687
START_RANGE client70a socket2236 630 139687-139750
START_RANGE client70a socket2237 620 139750-139812
START_RANGE client70a socket2238 630 139812-139875
START_RANGE client70a socket2239 620 139875-139937
START_RANGE client70a socket2240 630 139937-140000
START_RANGE client71a socket2241 620 140000-140062
START_RANGE client71a socket2242 630 140062-140125
START_RANGE client71a socket2243 620 140125-140187
START_RANGE client71a socket2244 630 140187-140250
START_RANGE client71a socket2245 620 140250-140312
START_RANGE client71a socket2246 630 140312-140375
START_RANGE client71a socket2247 620 140375-140437
START_RANGE client71a socket2248 630 140437-140500
START_RANGE client71a socket2249 620 140500-140562
START_RANGE client71a socket2250 630 140562-140625
START_RANGE client71a socket2251 620 140625-140687
START_RANGE client71a socket2252 630 140687-140750
START_RANGE client71a socket2253 620 140750-140812
START_RANGE client71a socket2254 630 140812-140875
START_RANGE client71a socket2255 620 140875-140937
START_RANGE client71a socket2256 630 140937-141000
START_RANGE client71a socket2257 620 141000-141062
START_RANGE client71a socket2258 630 141062-141125
START_RANGE client71a socket2259 620 141125-141187
START_RANGE client71a socket2260 630 141187-141250
START_RANGE client71a socket2261 620 141250-141312
START_RANGE client71a socket2262 630 141312-141375
START_RANGE client71a socket2263 620 141375-141437
START_RANGE client71a socket2264 630 141437-141500
START_RANGE client71a socket2265 620 141500-141562
START_RANGE client71a socket2266 630 141562-141625
START_RANGE client71a socket2267 620 141625-141687
START_RANGE client71a socket2268 630 141687-141750
START_RANGE client71a socket2269 620 141750-141812
START_RANGE client71a socket2270 630 141812-141875
START_RANGE client71a socket2271 620 141875-141937
START_RANGE client71a socket2272 630 141937-142000
START_RANGE client72a socket2273 620 142000-142062
START_RANGE client72a socket2274 630 142062-142125
START_RANGE client72a socket2275 620 142125-142187
START_RANGE client72a socket2276 630 142187-142250
START_RANGE client72a socket2277 620 142250-142312
START_RANGE client72a socket2278 630 142312-142375
START_RANGE client72a socket2279 620 142375-142437
START_RANGE client72a socket2280 630 142437-142500
START_RANGE client72a socket2281 620 142500-142562
START_RANGE client72a socket2282 630 142562-142625
START_RANGE client72a socket2283 620 142625-142687
START_RANGE client72a socket2284 630 142687-142750
START_RANGE client72a socket2285 620 142750-142812
START_RANGE client72a socket2286 630 142812-142875

START_RANGE client72a socket2287 620 142875-142937
START_RANGE client72a socket2288 630 142937-143000
START_RANGE client72a socket2289 620 143000-143062
START_RANGE client72a socket2290 630 143062-143125
START_RANGE client72a socket2291 620 143125-143187
START_RANGE client72a socket2292 630 143187-143250
START_RANGE client72a socket2293 620 143250-143312
START_RANGE client72a socket2294 630 143312-143375
START_RANGE client72a socket2295 620 143375-143437
START_RANGE client72a socket2296 630 143437-143500
START_RANGE client72a socket2297 620 143500-143562
START_RANGE client72a socket2298 630 143562-143625
START_RANGE client72a socket2299 620 143625-143687
START_RANGE client72a socket2300 630 143687-143750
START_RANGE client72a socket2301 620 143750-143812
START_RANGE client72a socket2302 630 143812-143875
START_RANGE client72a socket2303 620 143875-143937
START_RANGE client72a socket2304 630 143937-144000
#elif MASTER_NUM19
START_RANGE client73a socket2305 620 144000-144062
START_RANGE client73a socket2306 630 144062-144125
START_RANGE client73a socket2307 620 144125-144187
START_RANGE client73a socket2308 630 144187-144250
START_RANGE client73a socket2309 620 144250-144312
START_RANGE client73a socket2310 630 144312-144375
START_RANGE client73a socket2311 620 144375-144437
START_RANGE client73a socket2312 630 144437-144500
START_RANGE client73a socket2313 620 144500-144562
START_RANGE client73a socket2314 630 144562-144625
START_RANGE client73a socket2315 620 144625-144687
START_RANGE client73a socket2316 630 144687-144750
START_RANGE client73a socket2317 620 144750-144812
START_RANGE client73a socket2318 630 144812-144875
START_RANGE client73a socket2319 620 144875-144937
START_RANGE client73a socket2320 630 144937-145000
START_RANGE client73a socket2321 620 145000-145062
START_RANGE client73a socket2322 630 145062-145125
START_RANGE client73a socket2323 620 145125-145187
START_RANGE client73a socket2324 630 145187-145250
START_RANGE client73a socket2325 620 145250-145312
START_RANGE client73a socket2326 630 145312-145375
START_RANGE client73a socket2327 620 145375-145437
START_RANGE client73a socket2328 630 145437-145500
START_RANGE client73a socket2329 620 145500-145562
START_RANGE client73a socket2330 630 145562-145625
START_RANGE client73a socket2331 620 145625-145687
START_RANGE client73a socket2332 630 145687-145750
START_RANGE client73a socket2333 620 145750-145812
START_RANGE client73a socket2334 630 145812-145875
START_RANGE client73a socket2335 620 145875-145937
START_RANGE client73a socket2336 630 145937-146000
START_RANGE client74a socket2337 620 146000-146062
START_RANGE client74a socket2338 630 146062-146125
START_RANGE client74a socket2339 620 146125-146187
START_RANGE client74a socket2340 630 146187-146250
START_RANGE client74a socket2341 620 146250-146312
START_RANGE client74a socket2342 630 146312-146375
START_RANGE client74a socket2343 620 146375-146437
START_RANGE client74a socket2344 630 146437-146500
START_RANGE client74a socket2345 620 146500-146562
START_RANGE client74a socket2346 630 146562-146625
START_RANGE client74a socket2347 620 146625-146687

START_RANGE client74a socket2348 630 146687-146750
START_RANGE client74a socket2349 620 146750-146812
START_RANGE client74a socket2350 630 146812-146875
START_RANGE client74a socket2351 620 146875-146937
START_RANGE client74a socket2352 630 146937-147000
START_RANGE client74a socket2353 620 147000-147062
START_RANGE client74a socket2354 630 147062-147125
START_RANGE client74a socket2355 620 147125-147187
START_RANGE client74a socket2356 630 147187-147250
START_RANGE client74a socket2357 620 147250-147312
START_RANGE client74a socket2358 630 147312-147375
START_RANGE client74a socket2359 620 147375-147437
START_RANGE client74a socket2360 630 147437-147500
START_RANGE client74a socket2361 620 147500-147562
START_RANGE client74a socket2362 630 147562-147625
START_RANGE client74a socket2363 620 147625-147687
START_RANGE client74a socket2364 630 147687-147750
START_RANGE client74a socket2365 620 147750-147812
START_RANGE client74a socket2366 630 147812-147875
START_RANGE client74a socket2367 620 147875-147937
START_RANGE client74a socket2368 630 147937-148000
START_RANGE client75a socket2369 620 148000-148062
START_RANGE client75a socket2370 630 148062-148125
START_RANGE client75a socket2371 620 148125-148187
START_RANGE client75a socket2372 630 148187-148250
START_RANGE client75a socket2373 620 148250-148312
START_RANGE client75a socket2374 630 148312-148375
START_RANGE client75a socket2375 620 148375-148437
START_RANGE client75a socket2376 630 148437-148500
START_RANGE client75a socket2377 620 148500-148562
START_RANGE client75a socket2378 630 148562-148625
START_RANGE client75a socket2379 620 148625-148687
START_RANGE client75a socket2380 630 148687-148750
START_RANGE client75a socket2381 620 148750-148812
START_RANGE client75a socket2382 630 148812-148875
START_RANGE client75a socket2383 620 148875-148937
START_RANGE client75a socket2384 630 148937-149000
START_RANGE client75a socket2385 620 149000-149062
START_RANGE client75a socket2386 630 149062-149125
START_RANGE client75a socket2387 620 149125-149187
START_RANGE client75a socket2388 630 149187-149250
START_RANGE client75a socket2389 620 149250-149312
START_RANGE client75a socket2390 630 149312-149375
START_RANGE client75a socket2391 620 149375-149437
START_RANGE client75a socket2392 630 149437-149500
START_RANGE client75a socket2393 620 149500-149562
START_RANGE client75a socket2394 630 149562-149625
START_RANGE client75a socket2395 620 149625-149687
START_RANGE client75a socket2396 630 149687-149750
START_RANGE client75a socket2397 620 149750-149812
START_RANGE client75a socket2398 630 149812-149875
START_RANGE client75a socket2399 620 149875-149937
START_RANGE client75a socket2400 630 149937-150000
START_RANGE client76a socket2401 620 150000-150062
START_RANGE client76a socket2402 630 150062-150125
START_RANGE client76a socket2403 620 150125-150187
START_RANGE client76a socket2404 630 150187-150250
START_RANGE client76a socket2405 620 150250-150312
START_RANGE client76a socket2406 630 150312-150375
START_RANGE client76a socket2407 620 150375-150437
START_RANGE client76a socket2408 630 150437-150500
START_RANGE client76a socket2409 620 150500-150562

START_RANGE client76a socket2410 630 150562-150625
START_RANGE client76a socket2411 620 150625-150687
START_RANGE client76a socket2412 630 150687-150750
START_RANGE client76a socket2413 620 150750-150812
START_RANGE client76a socket2414 630 150812-150875
START_RANGE client76a socket2415 620 150875-150937
START_RANGE client76a socket2416 630 150937-151000
START_RANGE client76a socket2417 620 151000-151062
START_RANGE client76a socket2418 630 151062-151125
START_RANGE client76a socket2419 620 151125-151187
START_RANGE client76a socket2420 630 151187-151250
START_RANGE client76a socket2421 620 151250-151312
START_RANGE client76a socket2422 630 151312-151375
START_RANGE client76a socket2423 620 151375-151437
START_RANGE client76a socket2424 630 151437-151500
START_RANGE client76a socket2425 620 151500-151562
START_RANGE client76a socket2426 630 151562-151625
START_RANGE client76a socket2427 620 151625-151687
START_RANGE client76a socket2428 630 151687-151750
START_RANGE client76a socket2429 620 151750-151812
START_RANGE client76a socket2430 630 151812-151875
START_RANGE client76a socket2431 620 151875-151937
START_RANGE client76a socket2432 630 151937-152000
#elif MASTER_NUM20
START_RANGE client77a socket2433 620 152000-152062
START_RANGE client77a socket2434 630 152062-152125
START_RANGE client77a socket2435 620 152125-152187
START_RANGE client77a socket2436 630 152187-152250
START_RANGE client77a socket2437 620 152250-152312
START_RANGE client77a socket2438 630 152312-152375
START_RANGE client77a socket2439 620 152375-152437
START_RANGE client77a socket2440 630 152437-152500
START_RANGE client77a socket2441 620 152500-152562
START_RANGE client77a socket2442 630 152562-152625
START_RANGE client77a socket2443 620 152625-152687
START_RANGE client77a socket2444 630 152687-152750
START_RANGE client77a socket2445 620 152750-152812
START_RANGE client77a socket2446 630 152812-152875
START_RANGE client77a socket2447 620 152875-152937
START_RANGE client77a socket2448 630 152937-153000
START_RANGE client77a socket2449 620 153000-153062
START_RANGE client77a socket2450 630 153062-153125
START_RANGE client77a socket2451 620 153125-153187
START_RANGE client77a socket2452 630 153187-153250
START_RANGE client77a socket2453 620 153250-153312
START_RANGE client77a socket2454 630 153312-153375
START_RANGE client77a socket2455 620 153375-153437
START_RANGE client77a socket2456 630 153437-153500
START_RANGE client77a socket2457 620 153500-153562
START_RANGE client77a socket2458 630 153562-153625
START_RANGE client77a socket2459 620 153625-153687
START_RANGE client77a socket2460 630 153687-153750
START_RANGE client77a socket2461 620 153750-153812
START_RANGE client77a socket2462 630 153812-153875
START_RANGE client77a socket2463 620 153875-153937
START_RANGE client77a socket2464 630 153937-154000
START_RANGE client78a socket2465 620 154000-154062
START_RANGE client78a socket2466 630 154062-154125
START_RANGE client78a socket2467 620 154125-154187
START_RANGE client78a socket2468 630 154187-154250
START_RANGE client78a socket2469 620 154250-154312
START_RANGE client78a socket2470 630 154312-154375

START_RANGE client78a socket2471 620 154375-154437
START_RANGE client78a socket2472 630 154437-154500
START_RANGE client78a socket2473 620 154500-154562
START_RANGE client78a socket2474 630 154562-154625
START_RANGE client78a socket2475 620 154625-154687
START_RANGE client78a socket2476 630 154687-154750
START_RANGE client78a socket2477 620 154750-154812
START_RANGE client78a socket2478 630 154812-154875
START_RANGE client78a socket2479 620 154875-154937
START_RANGE client78a socket2480 630 154937-155000
START_RANGE client78a socket2481 620 155000-155062
START_RANGE client78a socket2482 630 155062-155125
START_RANGE client78a socket2483 620 155125-155187
START_RANGE client78a socket2484 630 155187-155250
START_RANGE client78a socket2485 620 155250-155312
START_RANGE client78a socket2486 630 155312-155375
START_RANGE client78a socket2487 620 155375-155437
START_RANGE client78a socket2488 630 155437-155500
START_RANGE client78a socket2489 620 155500-155562
START_RANGE client78a socket2490 630 155562-155625
START_RANGE client78a socket2491 620 155625-155687
START_RANGE client78a socket2492 630 155687-155750
START_RANGE client78a socket2493 620 155750-155812
START_RANGE client78a socket2494 630 155812-155875
START_RANGE client78a socket2495 620 155875-155937
START_RANGE client78a socket2496 630 155937-156000
START_RANGE client79a socket2497 620 156000-156062
START_RANGE client79a socket2498 630 156062-156125
START_RANGE client79a socket2499 620 156125-156187
START_RANGE client79a socket2500 630 156187-156250
START_RANGE client79a socket2501 620 156250-156312
START_RANGE client79a socket2502 630 156312-156375
START_RANGE client79a socket2503 620 156375-156437
START_RANGE client79a socket2504 630 156437-156500
START_RANGE client79a socket2505 620 156500-156562
START_RANGE client79a socket2506 630 156562-156625
START_RANGE client79a socket2507 620 156625-156687
START_RANGE client79a socket2508 630 156687-156750
START_RANGE client79a socket2509 620 156750-156812
START_RANGE client79a socket2510 630 156812-156875
START_RANGE client79a socket2511 620 156875-156937
START_RANGE client79a socket2512 630 156937-157000
START_RANGE client79a socket2513 620 157000-157062
START_RANGE client79a socket2514 630 157062-157125
START_RANGE client79a socket2515 620 157125-157187
START_RANGE client79a socket2516 630 157187-157250
START_RANGE client79a socket2517 620 157250-157312
START_RANGE client79a socket2518 630 157312-157375
START_RANGE client79a socket2519 620 157375-157437
START_RANGE client79a socket2520 630 157437-157500
START_RANGE client79a socket2521 620 157500-157562
START_RANGE client79a socket2522 630 157562-157625
START_RANGE client79a socket2523 620 157625-157687
START_RANGE client79a socket2524 630 157687-157750
START_RANGE client79a socket2525 620 157750-157812
START_RANGE client79a socket2526 630 157812-157875
START_RANGE client79a socket2527 620 157875-157937
START_RANGE client79a socket2528 630 157937-158000
START_RANGE client80a socket2529 620 158000-158062
START_RANGE client80a socket2530 630 158062-158125
START_RANGE client80a socket2531 620 158125-158187
START_RANGE client80a socket2532 630 158187-158250

START_RANGE client80a socket2533 620 158250-158312
START_RANGE client80a socket2534 630 158312-158375
START_RANGE client80a socket2535 620 158375-158437
START_RANGE client80a socket2536 630 158437-158500
START_RANGE client80a socket2537 620 158500-158562
START_RANGE client80a socket2538 630 158562-158625
START_RANGE client80a socket2539 620 158625-158687
START_RANGE client80a socket2540 630 158687-158750
START_RANGE client80a socket2541 620 158750-158812
START_RANGE client80a socket2542 630 158812-158875
START_RANGE client80a socket2543 620 158875-158937
START_RANGE client80a socket2544 630 158937-159000
START_RANGE client80a socket2545 620 159000-159062
START_RANGE client80a socket2546 630 159062-159125
START_RANGE client80a socket2547 620 159125-159187
START_RANGE client80a socket2548 630 159187-159250
START_RANGE client80a socket2549 620 159250-159312
START_RANGE client80a socket2550 630 159312-159375
START_RANGE client80a socket2551 620 159375-159437
START_RANGE client80a socket2552 630 159437-159500
START_RANGE client80a socket2553 620 159500-159562
START_RANGE client80a socket2554 630 159562-159625
START_RANGE client80a socket2555 620 159625-159687
START_RANGE client80a socket2556 630 159687-159750
START_RANGE client80a socket2557 620 159750-159812
START_RANGE client80a socket2558 630 159812-159875
START_RANGE client80a socket2559 620 159875-159937
START_RANGE client80a socket2560 630 159937-160000
#elif MASTER_NUM21
START_RANGE client81a socket2561 620 160000-160062
START_RANGE client81a socket2562 630 160062-160125
START_RANGE client81a socket2563 620 160125-160187
START_RANGE client81a socket2564 630 160187-160250
START_RANGE client81a socket2565 620 160250-160312
START_RANGE client81a socket2566 630 160312-160375
START_RANGE client81a socket2567 620 160375-160437
START_RANGE client81a socket2568 630 160437-160500
START_RANGE client81a socket2569 620 160500-160562
START_RANGE client81a socket2570 630 160562-160625
START_RANGE client81a socket2571 620 160625-160687
START_RANGE client81a socket2572 630 160687-160750
START_RANGE client81a socket2573 620 160750-160812
START_RANGE client81a socket2574 630 160812-160875
START_RANGE client81a socket2575 620 160875-160937
START_RANGE client81a socket2576 630 160937-161000
START_RANGE client81a socket2577 620 161000-161062
START_RANGE client81a socket2578 630 161062-161125
START_RANGE client81a socket2579 620 161125-161187
START_RANGE client81a socket2580 630 161187-161250
START_RANGE client81a socket2581 620 161250-161312
START_RANGE client81a socket2582 630 161312-161375
START_RANGE client81a socket2583 620 161375-161437
START_RANGE client81a socket2584 630 161437-161500
START_RANGE client81a socket2585 620 161500-161562
START_RANGE client81a socket2586 630 161562-161625
START_RANGE client81a socket2587 620 161625-161687
START_RANGE client81a socket2588 630 161687-161750
START_RANGE client81a socket2589 620 161750-161812
START_RANGE client81a socket2590 630 161812-161875
START_RANGE client81a socket2591 620 161875-161937
START_RANGE client81a socket2592 630 161937-162000
START_RANGE client82a socket2593 620 162000-162062

START_RANGE client82a socket2594 630 162062-162125
START_RANGE client82a socket2595 620 162125-162187
START_RANGE client82a socket2596 630 162187-162250
START_RANGE client82a socket2597 620 162250-162312
START_RANGE client82a socket2598 630 162312-162375
START_RANGE client82a socket2599 620 162375-162437
START_RANGE client82a socket2600 630 162437-162500
START_RANGE client82a socket2601 620 162500-162562
START_RANGE client82a socket2602 630 162562-162625
START_RANGE client82a socket2603 620 162625-162687
START_RANGE client82a socket2604 630 162687-162750
START_RANGE client82a socket2605 620 162750-162812
START_RANGE client82a socket2606 630 162812-162875
START_RANGE client82a socket2607 620 162875-162937
START_RANGE client82a socket2608 630 162937-163000
START_RANGE client82a socket2609 620 163000-163062
START_RANGE client82a socket2610 630 163062-163125
START_RANGE client82a socket2611 620 163125-163187
START_RANGE client82a socket2612 630 163187-163250
START_RANGE client82a socket2613 620 163250-163312
START_RANGE client82a socket2614 630 163312-163375
START_RANGE client82a socket2615 620 163375-163437
START_RANGE client82a socket2616 630 163437-163500
START_RANGE client82a socket2617 620 163500-163562
START_RANGE client82a socket2618 630 163562-163625
START_RANGE client82a socket2619 620 163625-163687
START_RANGE client82a socket2620 630 163687-163750
START_RANGE client82a socket2621 620 163750-163812
START_RANGE client82a socket2622 630 163812-163875
START_RANGE client82a socket2623 620 163875-163937
START_RANGE client82a socket2624 630 163937-164000
START_RANGE client83a socket2625 620 164000-164062
START_RANGE client83a socket2626 630 164062-164125
START_RANGE client83a socket2627 620 164125-164187
START_RANGE client83a socket2628 630 164187-164250
START_RANGE client83a socket2629 620 164250-164312
START_RANGE client83a socket2630 630 164312-164375
START_RANGE client83a socket2631 620 164375-164437
START_RANGE client83a socket2632 630 164437-164500
START_RANGE client83a socket2633 620 164500-164562
START_RANGE client83a socket2634 630 164562-164625
START_RANGE client83a socket2635 620 164625-164687
START_RANGE client83a socket2636 630 164687-164750
START_RANGE client83a socket2637 620 164750-164812
START_RANGE client83a socket2638 630 164812-164875
START_RANGE client83a socket2639 620 164875-164937
START_RANGE client83a socket2640 630 164937-165000
START_RANGE client83a socket2641 620 165000-165062
START_RANGE client83a socket2642 630 165062-165125
START_RANGE client83a socket2643 620 165125-165187
START_RANGE client83a socket2644 630 165187-165250
START_RANGE client83a socket2645 620 165250-165312
START_RANGE client83a socket2646 630 165312-165375
START_RANGE client83a socket2647 620 165375-165437
START_RANGE client83a socket2648 630 165437-165500
START_RANGE client83a socket2649 620 165500-165562
START_RANGE client83a socket2650 630 165562-165625
START_RANGE client83a socket2651 620 165625-165687
START_RANGE client83a socket2652 630 165687-165750
START_RANGE client83a socket2653 620 165750-165812
START_RANGE client83a socket2654 630 165812-165875
START_RANGE client83a socket2655 620 165875-165937

START_RANGE client83a socket2656 630 165937-166000
START_RANGE client84a socket2657 620 166000-166062
START_RANGE client84a socket2658 630 166062-166125
START_RANGE client84a socket2659 620 166125-166187
START_RANGE client84a socket2660 630 166187-166250
START_RANGE client84a socket2661 620 166250-166312
START_RANGE client84a socket2662 630 166312-166375
START_RANGE client84a socket2663 620 166375-166437
START_RANGE client84a socket2664 630 166437-166500
START_RANGE client84a socket2665 620 166500-166562
START_RANGE client84a socket2666 630 166562-166625
START_RANGE client84a socket2667 620 166625-166687
START_RANGE client84a socket2668 630 166687-166750
START_RANGE client84a socket2669 620 166750-166812
START_RANGE client84a socket2670 630 166812-166875
START_RANGE client84a socket2671 620 166875-166937
START_RANGE client84a socket2672 630 166937-167000
START_RANGE client84a socket2673 620 167000-167062
START_RANGE client84a socket2674 630 167062-167125
START_RANGE client84a socket2675 620 167125-167187
START_RANGE client84a socket2676 630 167187-167250
START_RANGE client84a socket2677 620 167250-167312
START_RANGE client84a socket2678 630 167312-167375
START_RANGE client84a socket2679 620 167375-167437
START_RANGE client84a socket2680 630 167437-167500
START_RANGE client84a socket2681 620 167500-167562
START_RANGE client84a socket2682 630 167562-167625
START_RANGE client84a socket2683 620 167625-167687
START_RANGE client84a socket2684 630 167687-167750
START_RANGE client84a socket2685 620 167750-167812
START_RANGE client84a socket2686 630 167812-167875
START_RANGE client84a socket2687 620 167875-167937
START_RANGE client84a socket2688 630 167937-168000
#elif MASTER_NUM22
START_RANGE client85a socket2689 620 168000-168062
START_RANGE client85a socket2690 630 168062-168125
START_RANGE client85a socket2691 620 168125-168187
START_RANGE client85a socket2692 630 168187-168250
START_RANGE client85a socket2693 620 168250-168312
START_RANGE client85a socket2694 630 168312-168375
START_RANGE client85a socket2695 620 168375-168437
START_RANGE client85a socket2696 630 168437-168500
START_RANGE client85a socket2697 620 168500-168562
START_RANGE client85a socket2698 630 168562-168625
START_RANGE client85a socket2699 620 168625-168687
START_RANGE client85a socket2700 630 168687-168750
START_RANGE client85a socket2701 620 168750-168812
START_RANGE client85a socket2702 630 168812-168875
START_RANGE client85a socket2703 620 168875-168937
START_RANGE client85a socket2704 630 168937-169000
START_RANGE client85a socket2705 620 169000-169062
START_RANGE client85a socket2706 630 169062-169125
START_RANGE client85a socket2707 620 169125-169187
START_RANGE client85a socket2708 630 169187-169250
START_RANGE client85a socket2709 620 169250-169312
START_RANGE client85a socket2710 630 169312-169375
START_RANGE client85a socket2711 620 169375-169437
START_RANGE client85a socket2712 630 169437-169500
START_RANGE client85a socket2713 620 169500-169562
START_RANGE client85a socket2714 630 169562-169625
START_RANGE client85a socket2715 620 169625-169687
START_RANGE client85a socket2716 630 169687-169750

START_RANGE client85a socket2717 620 169750-169812
START_RANGE client85a socket2718 630 169812-169875
START_RANGE client85a socket2719 620 169875-169937
START_RANGE client85a socket2720 630 169937-170000
START_RANGE client86a socket2721 620 170000-170062
START_RANGE client86a socket2722 630 170062-170125
START_RANGE client86a socket2723 620 170125-170187
START_RANGE client86a socket2724 630 170187-170250
START_RANGE client86a socket2725 620 170250-170312
START_RANGE client86a socket2726 630 170312-170375
START_RANGE client86a socket2727 620 170375-170437
START_RANGE client86a socket2728 630 170437-170500
START_RANGE client86a socket2729 620 170500-170562
START_RANGE client86a socket2730 630 170562-170625
START_RANGE client86a socket2731 620 170625-170687
START_RANGE client86a socket2732 630 170687-170750
START_RANGE client86a socket2733 620 170750-170812
START_RANGE client86a socket2734 630 170812-170875
START_RANGE client86a socket2735 620 170875-170937
START_RANGE client86a socket2736 630 170937-171000
START_RANGE client86a socket2737 620 171000-171062
START_RANGE client86a socket2738 630 171062-171125
START_RANGE client86a socket2739 620 171125-171187
START_RANGE client86a socket2740 630 171187-171250
START_RANGE client86a socket2741 620 171250-171312
START_RANGE client86a socket2742 630 171312-171375
START_RANGE client86a socket2743 620 171375-171437
START_RANGE client86a socket2744 630 171437-171500
START_RANGE client86a socket2745 620 171500-171562
START_RANGE client86a socket2746 630 171562-171625
START_RANGE client86a socket2747 620 171625-171687
START_RANGE client86a socket2748 630 171687-171750
START_RANGE client86a socket2749 620 171750-171812
START_RANGE client86a socket2750 630 171812-171875
START_RANGE client86a socket2751 620 171875-171937
START_RANGE client86a socket2752 630 171937-172000
START_RANGE client87a socket2753 620 172000-172062
START_RANGE client87a socket2754 630 172062-172125
START_RANGE client87a socket2755 620 172125-172187
START_RANGE client87a socket2756 630 172187-172250
START_RANGE client87a socket2757 620 172250-172312
START_RANGE client87a socket2758 630 172312-172375
START_RANGE client87a socket2759 620 172375-172437
START_RANGE client87a socket2760 630 172437-172500
START_RANGE client87a socket2761 620 172500-172562
START_RANGE client87a socket2762 630 172562-172625
START_RANGE client87a socket2763 620 172625-172687
START_RANGE client87a socket2764 630 172687-172750
START_RANGE client87a socket2765 620 172750-172812
START_RANGE client87a socket2766 630 172812-172875
START_RANGE client87a socket2767 620 172875-172937
START_RANGE client87a socket2768 630 172937-173000
START_RANGE client87a socket2769 620 173000-173062
START_RANGE client87a socket2770 630 173062-173125
START_RANGE client87a socket2771 620 173125-173187
START_RANGE client87a socket2772 630 173187-173250
START_RANGE client87a socket2773 620 173250-173312
START_RANGE client87a socket2774 630 173312-173375
START_RANGE client87a socket2775 620 173375-173437
START_RANGE client87a socket2776 630 173437-173500
START_RANGE client87a socket2777 620 173500-173562
START_RANGE client87a socket2778 630 173562-173625

START_RANGE client87a socket2779 620 173625-173687
START_RANGE client87a socket2780 630 173687-173750
START_RANGE client87a socket2781 620 173750-173812
START_RANGE client87a socket2782 630 173812-173875
START_RANGE client87a socket2783 620 173875-173937
START_RANGE client87a socket2784 630 173937-174000
START_RANGE client88a socket2785 620 174000-174062
START_RANGE client88a socket2786 630 174062-174125
START_RANGE client88a socket2787 620 174125-174187
START_RANGE client88a socket2788 630 174187-174250
START_RANGE client88a socket2789 620 174250-174312
START_RANGE client88a socket2790 630 174312-174375
START_RANGE client88a socket2791 620 174375-174437
START_RANGE client88a socket2792 630 174437-174500
START_RANGE client88a socket2793 620 174500-174562
START_RANGE client88a socket2794 630 174562-174625
START_RANGE client88a socket2795 620 174625-174687
START_RANGE client88a socket2796 630 174687-174750
START_RANGE client88a socket2797 620 174750-174812
START_RANGE client88a socket2798 630 174812-174875
START_RANGE client88a socket2799 620 174875-174937
START_RANGE client88a socket2800 630 174937-175000
START_RANGE client88a socket2801 620 175000-175062
START_RANGE client88a socket2802 630 175062-175125
START_RANGE client88a socket2803 620 175125-175187
START_RANGE client88a socket2804 630 175187-175250
START_RANGE client88a socket2805 620 175250-175312
START_RANGE client88a socket2806 630 175312-175375
START_RANGE client88a socket2807 620 175375-175437
START_RANGE client88a socket2808 630 175437-175500
START_RANGE client88a socket2809 620 175500-175562
START_RANGE client88a socket2810 630 175562-175625
START_RANGE client88a socket2811 620 175625-175687
START_RANGE client88a socket2812 630 175687-175750
START_RANGE client88a socket2813 620 175750-175812
START_RANGE client88a socket2814 630 175812-175875
START_RANGE client88a socket2815 620 175875-175937
START_RANGE client88a socket2816 630 175937-176000
#elif MASTER_NUM23
START_RANGE client89a socket2817 620 176000-176062
START_RANGE client89a socket2818 630 176062-176125
START_RANGE client89a socket2819 620 176125-176187
START_RANGE client89a socket2820 630 176187-176250
START_RANGE client89a socket2821 620 176250-176312
START_RANGE client89a socket2822 630 176312-176375
START_RANGE client89a socket2823 620 176375-176437
START_RANGE client89a socket2824 630 176437-176500
START_RANGE client89a socket2825 620 176500-176562
START_RANGE client89a socket2826 630 176562-176625
START_RANGE client89a socket2827 620 176625-176687
START_RANGE client89a socket2828 630 176687-176750
START_RANGE client89a socket2829 620 176750-176812
START_RANGE client89a socket2830 630 176812-176875
START_RANGE client89a socket2831 620 176875-176937
START_RANGE client89a socket2832 630 176937-177000
START_RANGE client89a socket2833 620 177000-177062
START_RANGE client89a socket2834 630 177062-177125
START_RANGE client89a socket2835 620 177125-177187
START_RANGE client89a socket2836 630 177187-177250
START_RANGE client89a socket2837 620 177250-177312
START_RANGE client89a socket2838 630 177312-177375
START_RANGE client89a socket2839 620 177375-177437

START_RANGE client89a socket2840 630 177437-177500
START_RANGE client89a socket2841 620 177500-177562
START_RANGE client89a socket2842 630 177562-177625
START_RANGE client89a socket2843 620 177625-177687
START_RANGE client89a socket2844 630 177687-177750
START_RANGE client89a socket2845 620 177750-177812
START_RANGE client89a socket2846 630 177812-177875
START_RANGE client89a socket2847 620 177875-177937
START_RANGE client89a socket2848 630 177937-178000
START_RANGE client90a socket2849 620 178000-178062
START_RANGE client90a socket2850 630 178062-178125
START_RANGE client90a socket2851 620 178125-178187
START_RANGE client90a socket2852 630 178187-178250
START_RANGE client90a socket2853 620 178250-178312
START_RANGE client90a socket2854 630 178312-178375
START_RANGE client90a socket2855 620 178375-178437
START_RANGE client90a socket2856 630 178437-178500
START_RANGE client90a socket2857 620 178500-178562
START_RANGE client90a socket2858 630 178562-178625
START_RANGE client90a socket2859 620 178625-178687
START_RANGE client90a socket2860 630 178687-178750
START_RANGE client90a socket2861 620 178750-178812
START_RANGE client90a socket2862 630 178812-178875
START_RANGE client90a socket2863 620 178875-178937
START_RANGE client90a socket2864 630 178937-179000
START_RANGE client90a socket2865 620 179000-179062
START_RANGE client90a socket2866 630 179062-179125
START_RANGE client90a socket2867 620 179125-179187
START_RANGE client90a socket2868 630 179187-179250
START_RANGE client90a socket2869 620 179250-179312
START_RANGE client90a socket2870 630 179312-179375
START_RANGE client90a socket2871 620 179375-179437
START_RANGE client90a socket2872 630 179437-179500
START_RANGE client90a socket2873 620 179500-179562
START_RANGE client90a socket2874 630 179562-179625
START_RANGE client90a socket2875 620 179625-179687
START_RANGE client90a socket2876 630 179687-179750
START_RANGE client90a socket2877 620 179750-179812
START_RANGE client90a socket2878 630 179812-179875
START_RANGE client90a socket2879 620 179875-179937
START_RANGE client90a socket2880 630 179937-180000
START_RANGE client91a socket2881 620 180000-180062
START_RANGE client91a socket2882 630 180062-180125
START_RANGE client91a socket2883 620 180125-180187
START_RANGE client91a socket2884 630 180187-180250
START_RANGE client91a socket2885 620 180250-180312
START_RANGE client91a socket2886 630 180312-180375
START_RANGE client91a socket2887 620 180375-180437
START_RANGE client91a socket2888 630 180437-180500
START_RANGE client91a socket2889 620 180500-180562
START_RANGE client91a socket2890 630 180562-180625
START_RANGE client91a socket2891 620 180625-180687
START_RANGE client91a socket2892 630 180687-180750
START_RANGE client91a socket2893 620 180750-180812
START_RANGE client91a socket2894 630 180812-180875
START_RANGE client91a socket2895 620 180875-180937
START_RANGE client91a socket2896 630 180937-181000
START_RANGE client91a socket2897 620 181000-181062
START_RANGE client91a socket2898 630 181062-181125
START_RANGE client91a socket2899 620 181125-181187
START_RANGE client91a socket2900 630 181187-181250
START_RANGE client91a socket2901 620 181250-181312

START_RANGE client91a socket2902 630 181312-181375
START_RANGE client91a socket2903 620 181375-181437
START_RANGE client91a socket2904 630 181437-181500
START_RANGE client91a socket2905 620 181500-181562
START_RANGE client91a socket2906 630 181562-181625
START_RANGE client91a socket2907 620 181625-181687
START_RANGE client91a socket2908 630 181687-181750
START_RANGE client91a socket2909 620 181750-181812
START_RANGE client91a socket2910 630 181812-181875
START_RANGE client91a socket2911 620 181875-181937
START_RANGE client91a socket2912 630 181937-182000
START_RANGE client92a socket2913 620 182000-182062
START_RANGE client92a socket2914 630 182062-182125
START_RANGE client92a socket2915 620 182125-182187
START_RANGE client92a socket2916 630 182187-182250
START_RANGE client92a socket2917 620 182250-182312
START_RANGE client92a socket2918 630 182312-182375
START_RANGE client92a socket2919 620 182375-182437
START_RANGE client92a socket2920 630 182437-182500
START_RANGE client92a socket2921 620 182500-182562
START_RANGE client92a socket2922 630 182562-182625
START_RANGE client92a socket2923 620 182625-182687
START_RANGE client92a socket2924 630 182687-182750
START_RANGE client92a socket2925 620 182750-182812
START_RANGE client92a socket2926 630 182812-182875
START_RANGE client92a socket2927 620 182875-182937
START_RANGE client92a socket2928 630 182937-183000
START_RANGE client92a socket2929 620 183000-183062
START_RANGE client92a socket2930 630 183062-183125
START_RANGE client92a socket2931 620 183125-183187
START_RANGE client92a socket2932 630 183187-183250
START_RANGE client92a socket2933 620 183250-183312
START_RANGE client92a socket2934 630 183312-183375
START_RANGE client92a socket2935 620 183375-183437
START_RANGE client92a socket2936 630 183437-183500
START_RANGE client92a socket2937 620 183500-183562
START_RANGE client92a socket2938 630 183562-183625
START_RANGE client92a socket2939 620 183625-183687
START_RANGE client92a socket2940 630 183687-183750
START_RANGE client92a socket2941 620 183750-183812
START_RANGE client92a socket2942 630 183812-183875
START_RANGE client92a socket2943 620 183875-183937
START_RANGE client92a socket2944 630 183937-184000
#elif MASTER_NUM24
START_RANGE client93a socket2945 620 184000-184062
START_RANGE client93a socket2946 630 184062-184125
START_RANGE client93a socket2947 620 184125-184187
START_RANGE client93a socket2948 630 184187-184250
START_RANGE client93a socket2949 620 184250-184312
START_RANGE client93a socket2950 630 184312-184375
START_RANGE client93a socket2951 620 184375-184437
START_RANGE client93a socket2952 630 184437-184500
START_RANGE client93a socket2953 620 184500-184562
START_RANGE client93a socket2954 630 184562-184625
START_RANGE client93a socket2955 620 184625-184687
START_RANGE client93a socket2956 630 184687-184750
START_RANGE client93a socket2957 620 184750-184812
START_RANGE client93a socket2958 630 184812-184875
START_RANGE client93a socket2959 620 184875-184937
START_RANGE client93a socket2960 630 184937-185000
START_RANGE client93a socket2961 620 185000-185062
START_RANGE client93a socket2962 630 185062-185125

START_RANGE client93a socket2963 620 185125-185187
START_RANGE client93a socket2964 630 185187-185250
START_RANGE client93a socket2965 620 185250-185312
START_RANGE client93a socket2966 630 185312-185375
START_RANGE client93a socket2967 620 185375-185437
START_RANGE client93a socket2968 630 185437-185500
START_RANGE client93a socket2969 620 185500-185562
START_RANGE client93a socket2970 630 185562-185625
START_RANGE client93a socket2971 620 185625-185687
START_RANGE client93a socket2972 630 185687-185750
START_RANGE client93a socket2973 620 185750-185812
START_RANGE client93a socket2974 630 185812-185875
START_RANGE client93a socket2975 620 185875-185937
START_RANGE client93a socket2976 630 185937-186000
START_RANGE client94a socket2977 620 186000-186062
START_RANGE client94a socket2978 630 186062-186125
START_RANGE client94a socket2979 620 186125-186187
START_RANGE client94a socket2980 630 186187-186250
START_RANGE client94a socket2981 620 186250-186312
START_RANGE client94a socket2982 630 186312-186375
START_RANGE client94a socket2983 620 186375-186437
START_RANGE client94a socket2984 630 186437-186500
START_RANGE client94a socket2985 620 186500-186562
START_RANGE client94a socket2986 630 186562-186625
START_RANGE client94a socket2987 620 186625-186687
START_RANGE client94a socket2988 630 186687-186750
START_RANGE client94a socket2989 620 186750-186812
START_RANGE client94a socket2990 630 186812-186875
START_RANGE client94a socket2991 620 186875-186937
START_RANGE client94a socket2992 630 186937-187000
START_RANGE client94a socket2993 620 187000-187062
START_RANGE client94a socket2994 630 187062-187125
START_RANGE client94a socket2995 620 187125-187187
START_RANGE client94a socket2996 630 187187-187250
START_RANGE client94a socket2997 620 187250-187312
START_RANGE client94a socket2998 630 187312-187375
START_RANGE client94a socket2999 620 187375-187437
START_RANGE client94a socket3000 630 187437-187500
START_RANGE client94a socket3001 620 187500-187562
START_RANGE client94a socket3002 630 187562-187625
START_RANGE client94a socket3003 620 187625-187687
START_RANGE client94a socket3004 630 187687-187750
START_RANGE client94a socket3005 620 187750-187812
START_RANGE client94a socket3006 630 187812-187875
START_RANGE client94a socket3007 620 187875-187937
START_RANGE client94a socket3008 630 187937-188000
START_RANGE client95a socket3009 620 188000-188062
START_RANGE client95a socket3010 630 188062-188125
START_RANGE client95a socket3011 620 188125-188187
START_RANGE client95a socket3012 630 188187-188250
START_RANGE client95a socket3013 620 188250-188312
START_RANGE client95a socket3014 630 188312-188375
START_RANGE client95a socket3015 620 188375-188437
START_RANGE client95a socket3016 630 188437-188500
START_RANGE client95a socket3017 620 188500-188562
START_RANGE client95a socket3018 630 188562-188625
START_RANGE client95a socket3019 620 188625-188687
START_RANGE client95a socket3020 630 188687-188750
START_RANGE client95a socket3021 620 188750-188812
START_RANGE client95a socket3022 630 188812-188875
START_RANGE client95a socket3023 620 188875-188937
START_RANGE client95a socket3024 630 188937-189000

START_RANGE client95a socket3025 620 189000-189062
START_RANGE client95a socket3026 630 189062-189125
START_RANGE client95a socket3027 620 189125-189187
START_RANGE client95a socket3028 630 189187-189250
START_RANGE client95a socket3029 620 189250-189312
START_RANGE client95a socket3030 630 189312-189375
START_RANGE client95a socket3031 620 189375-189437
START_RANGE client95a socket3032 630 189437-189500
START_RANGE client95a socket3033 620 189500-189562
START_RANGE client95a socket3034 630 189562-189625
START_RANGE client95a socket3035 620 189625-189687
START_RANGE client95a socket3036 630 189687-189750
START_RANGE client95a socket3037 620 189750-189812
START_RANGE client95a socket3038 630 189812-189875
START_RANGE client95a socket3039 620 189875-189937
START_RANGE client95a socket3040 630 189937-190000
START_RANGE client96a socket3041 620 190000-190062
START_RANGE client96a socket3042 630 190062-190125
START_RANGE client96a socket3043 620 190125-190187
START_RANGE client96a socket3044 630 190187-190250
START_RANGE client96a socket3045 620 190250-190312
START_RANGE client96a socket3046 630 190312-190375
START_RANGE client96a socket3047 620 190375-190437
START_RANGE client96a socket3048 630 190437-190500
START_RANGE client96a socket3049 620 190500-190562
START_RANGE client96a socket3050 630 190562-190625
START_RANGE client96a socket3051 620 190625-190687
START_RANGE client96a socket3052 630 190687-190750
START_RANGE client96a socket3053 620 190750-190812
START_RANGE client96a socket3054 630 190812-190875
START_RANGE client96a socket3055 620 190875-190937
START_RANGE client96a socket3056 630 190937-191000
START_RANGE client96a socket3057 620 191000-191062
START_RANGE client96a socket3058 630 191062-191125
START_RANGE client96a socket3059 620 191125-191187
START_RANGE client96a socket3060 630 191187-191250
START_RANGE client96a socket3061 620 191250-191312
START_RANGE client96a socket3062 630 191312-191375
START_RANGE client96a socket3063 620 191375-191437
START_RANGE client96a socket3064 630 191437-191500
START_RANGE client96a socket3065 620 191500-191562
START_RANGE client96a socket3066 630 191562-191625
START_RANGE client96a socket3067 620 191625-191687
START_RANGE client96a socket3068 630 191687-191750
START_RANGE client96a socket3069 620 191750-191812
START_RANGE client96a socket3070 630 191812-191875
START_RANGE client96a socket3071 620 191875-191937
START_RANGE client96a socket3072 630 191937-192000
#elif MASTER_NUM25
START_RANGE client97a socket3073 620 192000-192062
START_RANGE client97a socket3074 630 192062-192125
START_RANGE client97a socket3075 620 192125-192187
START_RANGE client97a socket3076 630 192187-192250
START_RANGE client97a socket3077 620 192250-192312
START_RANGE client97a socket3078 630 192312-192375
START_RANGE client97a socket3079 620 192375-192437
START_RANGE client97a socket3080 630 192437-192500
START_RANGE client97a socket3081 620 192500-192562
START_RANGE client97a socket3082 630 192562-192625
START_RANGE client97a socket3083 620 192625-192687
START_RANGE client97a socket3084 630 192687-192750
START_RANGE client97a socket3085 620 192750-192812

START_RANGE client97a socket3086 630 192812-192875
START_RANGE client97a socket3087 620 192875-192937
START_RANGE client97a socket3088 630 192937-193000
START_RANGE client97a socket3089 620 193000-193062
START_RANGE client97a socket3090 630 193062-193125
START_RANGE client97a socket3091 620 193125-193187
START_RANGE client97a socket3092 630 193187-193250
START_RANGE client97a socket3093 620 193250-193312
START_RANGE client97a socket3094 630 193312-193375
START_RANGE client97a socket3095 620 193375-193437
START_RANGE client97a socket3096 630 193437-193500
START_RANGE client97a socket3097 620 193500-193562
START_RANGE client97a socket3098 630 193562-193625
START_RANGE client97a socket3099 620 193625-193687
START_RANGE client97a socket3100 630 193687-193750
START_RANGE client97a socket3101 620 193750-193812
START_RANGE client97a socket3102 630 193812-193875
START_RANGE client97a socket3103 620 193875-193937
START_RANGE client97a socket3104 630 193937-194000
START_RANGE client98a socket3105 620 194000-194062
START_RANGE client98a socket3106 630 194062-194125
START_RANGE client98a socket3107 620 194125-194187
START_RANGE client98a socket3108 630 194187-194250
START_RANGE client98a socket3109 620 194250-194312
START_RANGE client98a socket3110 630 194312-194375
START_RANGE client98a socket3111 620 194375-194437
START_RANGE client98a socket3112 630 194437-194500
START_RANGE client98a socket3113 620 194500-194562
START_RANGE client98a socket3114 630 194562-194625
START_RANGE client98a socket3115 620 194625-194687
START_RANGE client98a socket3116 630 194687-194750
START_RANGE client98a socket3117 620 194750-194812
START_RANGE client98a socket3118 630 194812-194875
START_RANGE client98a socket3119 620 194875-194937
START_RANGE client98a socket3120 630 194937-195000
START_RANGE client98a socket3121 620 195000-195062
START_RANGE client98a socket3122 630 195062-195125
START_RANGE client98a socket3123 620 195125-195187
START_RANGE client98a socket3124 630 195187-195250
START_RANGE client98a socket3125 620 195250-195312
START_RANGE client98a socket3126 630 195312-195375
START_RANGE client98a socket3127 620 195375-195437
START_RANGE client98a socket3128 630 195437-195500
START_RANGE client98a socket3129 620 195500-195562
START_RANGE client98a socket3130 630 195562-195625
START_RANGE client98a socket3131 620 195625-195687
START_RANGE client98a socket3132 630 195687-195750
START_RANGE client98a socket3133 620 195750-195812
START_RANGE client98a socket3134 630 195812-195875
START_RANGE client98a socket3135 620 195875-195937
START_RANGE client98a socket3136 630 195937-196000
START_RANGE client99a socket3137 620 196000-196062
START_RANGE client99a socket3138 630 196062-196125
START_RANGE client99a socket3139 620 196125-196187
START_RANGE client99a socket3140 630 196187-196250
START_RANGE client99a socket3141 620 196250-196312
START_RANGE client99a socket3142 630 196312-196375
START_RANGE client99a socket3143 620 196375-196437
START_RANGE client99a socket3144 630 196437-196500
START_RANGE client99a socket3145 620 196500-196562
START_RANGE client99a socket3146 630 196562-196625
START_RANGE client99a socket3147 620 196625-196687

START_RANGE client99a socket3148 630 196687-196750
START_RANGE client99a socket3149 620 196750-196812
START_RANGE client99a socket3150 630 196812-196875
START_RANGE client99a socket3151 620 196875-196937
START_RANGE client99a socket3152 630 196937-197000
START_RANGE client99a socket3153 620 197000-197062
START_RANGE client99a socket3154 630 197062-197125
START_RANGE client99a socket3155 620 197125-197187
START_RANGE client99a socket3156 630 197187-197250
START_RANGE client99a socket3157 620 197250-197312
START_RANGE client99a socket3158 630 197312-197375
START_RANGE client99a socket3159 620 197375-197437
START_RANGE client99a socket3160 630 197437-197500
START_RANGE client99a socket3161 620 197500-197562
START_RANGE client99a socket3162 630 197562-197625
START_RANGE client99a socket3163 620 197625-197687
START_RANGE client99a socket3164 630 197687-197750
START_RANGE client99a socket3165 620 197750-197812
START_RANGE client99a socket3166 630 197812-197875
START_RANGE client99a socket3167 620 197875-197937
START_RANGE client99a socket3168 630 197937-198000
START_RANGE client100a socket3169 620 198000-198062
START_RANGE client100a socket3170 630 198062-198125
START_RANGE client100a socket3171 620 198125-198187
START_RANGE client100a socket3172 630 198187-198250
START_RANGE client100a socket3173 620 198250-198312
START_RANGE client100a socket3174 630 198312-198375
START_RANGE client100a socket3175 620 198375-198437
START_RANGE client100a socket3176 630 198437-198500
START_RANGE client100a socket3177 620 198500-198562
START_RANGE client100a socket3178 630 198562-198625
START_RANGE client100a socket3179 620 198625-198687
START_RANGE client100a socket3180 630 198687-198750
START_RANGE client100a socket3181 620 198750-198812
START_RANGE client100a socket3182 630 198812-198875
START_RANGE client100a socket3183 620 198875-198937
START_RANGE client100a socket3184 630 198937-199000
START_RANGE client100a socket3185 620 199000-199062
START_RANGE client100a socket3186 630 199062-199125
START_RANGE client100a socket3187 620 199125-199187
START_RANGE client100a socket3188 630 199187-199250
START_RANGE client100a socket3189 620 199250-199312
START_RANGE client100a socket3190 630 199312-199375
START_RANGE client100a socket3191 620 199375-199437
START_RANGE client100a socket3192 630 199437-199500
START_RANGE client100a socket3193 620 199500-199562
START_RANGE client100a socket3194 630 199562-199625
START_RANGE client100a socket3195 620 199625-199687
START_RANGE client100a socket3196 630 199687-199750
START_RANGE client100a socket3197 620 199750-199812
START_RANGE client100a socket3198 630 199812-199875
START_RANGE client100a socket3199 620 199875-199937
START_RANGE client100a socket3200 630 199937-200000
#elif MASTER_NUM26
START_RANGE client101a socket3201 620 200000-200062
START_RANGE client101a socket3202 630 200062-200125
START_RANGE client101a socket3203 620 200125-200187
START_RANGE client101a socket3204 630 200187-200250
START_RANGE client101a socket3205 620 200250-200312
START_RANGE client101a socket3206 630 200312-200375
START_RANGE client101a socket3207 620 200375-200437
START_RANGE client101a socket3208 630 200437-200500

START_RANGE client101a socket3209 620 200500-200562
START_RANGE client101a socket3210 630 200562-200625
START_RANGE client101a socket3211 620 200625-200687
START_RANGE client101a socket3212 630 200687-200750
START_RANGE client101a socket3213 620 200750-200812
START_RANGE client101a socket3214 630 200812-200875
START_RANGE client101a socket3215 620 200875-200937
START_RANGE client101a socket3216 630 200937-201000
START_RANGE client101a socket3217 620 201000-201062
START_RANGE client101a socket3218 630 201062-201125
START_RANGE client101a socket3219 620 201125-201187
START_RANGE client101a socket3220 630 201187-201250
START_RANGE client101a socket3221 620 201250-201312
START_RANGE client101a socket3222 630 201312-201375
START_RANGE client101a socket3223 620 201375-201437
START_RANGE client101a socket3224 630 201437-201500
START_RANGE client101a socket3225 620 201500-201562
START_RANGE client101a socket3226 630 201562-201625
START_RANGE client101a socket3227 620 201625-201687
START_RANGE client101a socket3228 630 201687-201750
START_RANGE client101a socket3229 620 201750-201812
START_RANGE client101a socket3230 630 201812-201875
START_RANGE client101a socket3231 620 201875-201937
START_RANGE client101a socket3232 630 201937-202000
START_RANGE client102a socket3233 620 202000-202062
START_RANGE client102a socket3234 630 202062-202125
START_RANGE client102a socket3235 620 202125-202187
START_RANGE client102a socket3236 630 202187-202250
START_RANGE client102a socket3237 620 202250-202312
START_RANGE client102a socket3238 630 202312-202375
START_RANGE client102a socket3239 620 202375-202437
START_RANGE client102a socket3240 630 202437-202500
START_RANGE client102a socket3241 620 202500-202562
START_RANGE client102a socket3242 630 202562-202625
START_RANGE client102a socket3243 620 202625-202687
START_RANGE client102a socket3244 630 202687-202750
START_RANGE client102a socket3245 620 202750-202812
START_RANGE client102a socket3246 630 202812-202875
START_RANGE client102a socket3247 620 202875-202937
START_RANGE client102a socket3248 630 202937-203000
START_RANGE client102a socket3249 620 203000-203062
START_RANGE client102a socket3250 630 203062-203125
START_RANGE client102a socket3251 620 203125-203187
START_RANGE client102a socket3252 630 203187-203250
START_RANGE client102a socket3253 620 203250-203312
START_RANGE client102a socket3254 630 203312-203375
START_RANGE client102a socket3255 620 203375-203437
START_RANGE client102a socket3256 630 203437-203500
START_RANGE client102a socket3257 620 203500-203562
START_RANGE client102a socket3258 630 203562-203625
START_RANGE client102a socket3259 620 203625-203687
START_RANGE client102a socket3260 630 203687-203750
START_RANGE client102a socket3261 620 203750-203812
START_RANGE client102a socket3262 630 203812-203875
START_RANGE client102a socket3263 620 203875-203937
START_RANGE client102a socket3264 630 203937-204000
START_RANGE client103a socket3265 620 204000-204062
START_RANGE client103a socket3266 630 204062-204125
START_RANGE client103a socket3267 620 204125-204187
START_RANGE client103a socket3268 630 204187-204250
START_RANGE client103a socket3269 620 204250-204312
START_RANGE client103a socket3270 630 204312-204375

START_RANGE client114a socket3640 630 227437-227500
START_RANGE client114a socket3641 620 227500-227562
START_RANGE client114a socket3642 630 227562-227625
START_RANGE client114a socket3643 620 227625-227687
START_RANGE client114a socket3644 630 227687-227750
START_RANGE client114a socket3645 620 227750-227812
START_RANGE client114a socket3646 630 227812-227875
START_RANGE client114a socket3647 620 227875-227937
START_RANGE client114a socket3648 630 227937-228000
START_RANGE client115a socket3649 620 228000-228062
START_RANGE client115a socket3650 630 228062-228125
START_RANGE client115a socket3651 620 228125-228187
START_RANGE client115a socket3652 630 228187-228250
START_RANGE client115a socket3653 620 228250-228312
START_RANGE client115a socket3654 630 228312-228375
START_RANGE client115a socket3655 620 228375-228437
START_RANGE client115a socket3656 630 228437-228500
START_RANGE client115a socket3657 620 228500-228562
START_RANGE client115a socket3658 630 228562-228625
START_RANGE client115a socket3659 620 228625-228687
START_RANGE client115a socket3660 630 228687-228750
START_RANGE client115a socket3661 620 228750-228812
START_RANGE client115a socket3662 630 228812-228875
START_RANGE client115a socket3663 620 228875-228937
START_RANGE client115a socket3664 630 228937-229000
START_RANGE client115a socket3665 620 229000-229062
START_RANGE client115a socket3666 630 229062-229125
START_RANGE client115a socket3667 620 229125-229187
START_RANGE client115a socket3668 630 229187-229250
START_RANGE client115a socket3669 620 229250-229312
START_RANGE client115a socket3670 630 229312-229375
START_RANGE client115a socket3671 620 229375-229437
START_RANGE client115a socket3672 630 229437-229500
START_RANGE client115a socket3673 620 229500-229562
START_RANGE client115a socket3674 630 229562-229625
START_RANGE client115a socket3675 620 229625-229687
START_RANGE client115a socket3676 630 229687-229750
START_RANGE client115a socket3677 620 229750-229812
START_RANGE client115a socket3678 630 229812-229875
START_RANGE client115a socket3679 620 229875-229937
START_RANGE client115a socket3680 630 229937-230000
START_RANGE client116a socket3681 620 230000-230062
START_RANGE client116a socket3682 630 230062-230125
START_RANGE client116a socket3683 620 230125-230187
START_RANGE client116a socket3684 630 230187-230250
START_RANGE client116a socket3685 620 230250-230312
START_RANGE client116a socket3686 630 230312-230375
START_RANGE client116a socket3687 620 230375-230437
START_RANGE client116a socket3688 630 230437-230500
START_RANGE client116a socket3689 620 230500-230562
START_RANGE client116a socket3690 630 230562-230625
START_RANGE client116a socket3691 620 230625-230687
START_RANGE client116a socket3692 630 230687-230750
START_RANGE client116a socket3693 620 230750-230812
START_RANGE client116a socket3694 630 230812-230875
START_RANGE client116a socket3695 620 230875-230937
START_RANGE client116a socket3696 630 230937-231000
START_RANGE client116a socket3697 620 231000-231062
START_RANGE client116a socket3698 630 231062-231125
START_RANGE client116a socket3699 620 231125-231187
START_RANGE client116a socket3700 630 231187-231250
START_RANGE client116a socket3701 620 231250-231312

START_RANGE client116a socket3702 630 231312-231375
START_RANGE client116a socket3703 620 231375-231437
START_RANGE client116a socket3704 630 231437-231500
START_RANGE client116a socket3705 620 231500-231562
START_RANGE client116a socket3706 630 231562-231625
START_RANGE client116a socket3707 620 231625-231687
START_RANGE client116a socket3708 630 231687-231750
START_RANGE client116a socket3709 620 231750-231812
START_RANGE client116a socket3710 630 231812-231875
START_RANGE client116a socket3711 620 231875-231937
START_RANGE client116a socket3712 630 231937-232000
#elif MASTER_NUM30
START_RANGE client117a socket3713 620 232000-232062
START_RANGE client117a socket3714 630 232062-232125
START_RANGE client117a socket3715 620 232125-232187
START_RANGE client117a socket3716 630 232187-232250
START_RANGE client117a socket3717 620 232250-232312
START_RANGE client117a socket3718 630 232312-232375
START_RANGE client117a socket3719 620 232375-232437
START_RANGE client117a socket3720 630 232437-232500
START_RANGE client117a socket3721 620 232500-232562
START_RANGE client117a socket3722 630 232562-232625
START_RANGE client117a socket3723 620 232625-232687
START_RANGE client117a socket3724 630 232687-232750
START_RANGE client117a socket3725 620 232750-232812
START_RANGE client117a socket3726 630 232812-232875
START_RANGE client117a socket3727 620 232875-232937
START_RANGE client117a socket3728 630 232937-233000
START_RANGE client117a socket3729 620 233000-233062
START_RANGE client117a socket3730 630 233062-233125
START_RANGE client117a socket3731 620 233125-233187
START_RANGE client117a socket3732 630 233187-233250
START_RANGE client117a socket3733 620 233250-233312
START_RANGE client117a socket3734 630 233312-233375
START_RANGE client117a socket3735 620 233375-233437
START_RANGE client117a socket3736 630 233437-233500
START_RANGE client117a socket3737 620 233500-233562
START_RANGE client117a socket3738 630 233562-233625
START_RANGE client117a socket3739 620 233625-233687
START_RANGE client117a socket3740 630 233687-233750
START_RANGE client117a socket3741 620 233750-233812
START_RANGE client117a socket3742 630 233812-233875
START_RANGE client117a socket3743 620 233875-233937
START_RANGE client117a socket3744 630 233937-234000
START_RANGE client118a socket3745 620 234000-234062
START_RANGE client118a socket3746 630 234062-234125
START_RANGE client118a socket3747 620 234125-234187
START_RANGE client118a socket3748 630 234187-234250
START_RANGE client118a socket3749 620 234250-234312
START_RANGE client118a socket3750 630 234312-234375
START_RANGE client118a socket3751 620 234375-234437
START_RANGE client118a socket3752 630 234437-234500
START_RANGE client118a socket3753 620 234500-234562
START_RANGE client118a socket3754 630 234562-234625
START_RANGE client118a socket3755 620 234625-234687
START_RANGE client118a socket3756 630 234687-234750
START_RANGE client118a socket3757 620 234750-234812
START_RANGE client118a socket3758 630 234812-234875
START_RANGE client118a socket3759 620 234875-234937
START_RANGE client118a socket3760 630 234937-235000
START_RANGE client118a socket3761 620 235000-235062
START_RANGE client118a socket3762 630 235062-235125

START_RANGE client118a socket3763 620 235125-235187
START_RANGE client118a socket3764 630 235187-235250
START_RANGE client118a socket3765 620 235250-235312
START_RANGE client118a socket3766 630 235312-235375
START_RANGE client118a socket3767 620 235375-235437
START_RANGE client118a socket3768 630 235437-235500
START_RANGE client118a socket3769 620 235500-235562
START_RANGE client118a socket3770 630 235562-235625
START_RANGE client118a socket3771 620 235625-235687
START_RANGE client118a socket3772 630 235687-235750
START_RANGE client118a socket3773 620 235750-235812
START_RANGE client118a socket3774 630 235812-235875
START_RANGE client118a socket3775 620 235875-235937
START_RANGE client118a socket3776 630 235937-236000
START_RANGE client119a socket3777 620 236000-236062
START_RANGE client119a socket3778 630 236062-236125
START_RANGE client119a socket3779 620 236125-236187
START_RANGE client119a socket3780 630 236187-236250
START_RANGE client119a socket3781 620 236250-236312
START_RANGE client119a socket3782 630 236312-236375
START_RANGE client119a socket3783 620 236375-236437
START_RANGE client119a socket3784 630 236437-236500
START_RANGE client119a socket3785 620 236500-236562
START_RANGE client119a socket3786 630 236562-236625
START_RANGE client119a socket3787 620 236625-236687
START_RANGE client119a socket3788 630 236687-236750
START_RANGE client119a socket3789 620 236750-236812
START_RANGE client119a socket3790 630 236812-236875
START_RANGE client119a socket3791 620 236875-236937
START_RANGE client119a socket3792 630 236937-237000
START_RANGE client119a socket3793 620 237000-237062
START_RANGE client119a socket3794 630 237062-237125
START_RANGE client119a socket3795 620 237125-237187
START_RANGE client119a socket3796 630 237187-237250
START_RANGE client119a socket3797 620 237250-237312
START_RANGE client119a socket3798 630 237312-237375
START_RANGE client119a socket3799 620 237375-237437
START_RANGE client119a socket3800 630 237437-237500
START_RANGE client119a socket3801 620 237500-237562
START_RANGE client119a socket3802 630 237562-237625
START_RANGE client119a socket3803 620 237625-237687
START_RANGE client119a socket3804 630 237687-237750
START_RANGE client119a socket3805 620 237750-237812
START_RANGE client119a socket3806 630 237812-237875
START_RANGE client119a socket3807 620 237875-237937
START_RANGE client119a socket3808 630 237937-238000
START_RANGE client120a socket3809 620 238000-238062
START_RANGE client120a socket3810 630 238062-238125
START_RANGE client120a socket3811 620 238125-238187
START_RANGE client120a socket3812 630 238187-238250
START_RANGE client120a socket3813 620 238250-238312
START_RANGE client120a socket3814 630 238312-238375
START_RANGE client120a socket3815 620 238375-238437
START_RANGE client120a socket3816 630 238437-238500
START_RANGE client120a socket3817 620 238500-238562
START_RANGE client120a socket3818 630 238562-238625
START_RANGE client120a socket3819 620 238625-238687
START_RANGE client120a socket3820 630 238687-238750
START_RANGE client120a socket3821 620 238750-238812
START_RANGE client120a socket3822 630 238812-238875
START_RANGE client120a socket3823 620 238875-238937
START_RANGE client120a socket3824 630 238937-239000


```

START_RANGE client126a socket4009 620 250500-250562
START_RANGE client126a socket4010 630 250562-250625
START_RANGE client126a socket4011 620 250625-250687
START_RANGE client126a socket4012 630 250687-250750
START_RANGE client126a socket4013 620 250750-250812
START_RANGE client126a socket4014 630 250812-250875
START_RANGE client126a socket4015 620 250875-250937
START_RANGE client126a socket4016 630 250937-251000
START_RANGE client126a socket4017 620 251000-251062
START_RANGE client126a socket4018 630 251062-251125
START_RANGE client126a socket4019 620 251125-251187
START_RANGE client126a socket4020 630 251187-251250
START_RANGE client126a socket4021 620 251250-251312
START_RANGE client126a socket4022 630 251312-251375
START_RANGE client126a socket4023 620 251375-251437
START_RANGE client126a socket4024 630 251437-251500
START_RANGE client126a socket4025 620 251500-251562
START_RANGE client126a socket4026 630 251562-251625
START_RANGE client126a socket4027 620 251625-251687
START_RANGE client126a socket4028 630 251687-251750
START_RANGE client126a socket4029 620 251750-251812
START_RANGE client126a socket4030 630 251812-251875
START_RANGE client126a socket4031 620 251875-251937
START_RANGE client126a socket4032 630 251937-252000
START_RANGE client127a socket4033 620 252000-252062
START_RANGE client127a socket4034 630 252062-252125
START_RANGE client127a socket4035 620 252125-252187
START_RANGE client127a socket4036 630 252187-252250
START_RANGE client127a socket4037 620 252250-252312
START_RANGE client127a socket4038 630 252312-252375
START_RANGE client127a socket4039 620 252375-252437
START_RANGE client127a socket4040 630 252437-252500
START_RANGE client127a socket4041 620 252500-252562
START_RANGE client127a socket4042 630 252562-252625
START_RANGE client127a socket4043 620 252625-252687
START_RANGE client127a socket4044 630 252687-252750
START_RANGE client127a socket4045 620 252750-252812
START_RANGE client127a socket4046 630 252812-252875
START_RANGE client127a socket4047 620 252875-252937
START_RANGE client127a socket4048 630 252937-253000
START_RANGE client127a socket4049 620 253000-253062
START_RANGE client127a socket4050 630 253062-253125
START_RANGE client127a socket4051 620 253125-253187
START_RANGE client127a socket4052 630 253187-253250
START_RANGE client127a socket4053 620 253250-253312
START_RANGE client127a socket4054 630 253312-253375
START_RANGE client127a socket4055 620 253375-253437
START_RANGE client127a socket4056 630 253437-253500
START_RANGE client127a socket4057 620 253500-253562
START_RANGE client127a socket4058 630 253562-253625
START_RANGE client127a socket4059 620 253625-253687
START_RANGE client127a socket4060 630 253687-253750
START_RANGE client127a socket4061 620 253750-253812
START_RANGE client127a socket4062 630 253812-253875
START_RANGE client127a socket4063 620 253875-253937
START_RANGE client127a socket4064 630 253937-254000
START_RANGE client128a socket4065 620 254000-254062
START_RANGE client128a socket4066 630 254062-254125
START_RANGE client128a socket4067 620 254125-254187
START_RANGE client128a socket4068 630 254187-254250
START_RANGE client128a socket4069 620 254250-254312
START_RANGE client128a socket4070 630 254312-254375

```

```

START_RANGE client128a socket4071 620 254375-254437
START_RANGE client128a socket4072 630 254437-254500
START_RANGE client128a socket4073 620 254500-254562
START_RANGE client128a socket4074 630 254562-254625
START_RANGE client128a socket4075 620 254625-254687
START_RANGE client128a socket4076 630 254687-254750
START_RANGE client128a socket4077 620 254750-254812
START_RANGE client128a socket4078 630 254812-254875
START_RANGE client128a socket4079 620 254875-254937
START_RANGE client128a socket4080 630 254937-255000
START_RANGE client128a socket4081 620 255000-255062
START_RANGE client128a socket4082 630 255062-255125
START_RANGE client128a socket4083 620 255125-255187
START_RANGE client128a socket4084 630 255187-255250
START_RANGE client128a socket4085 620 255250-255312
START_RANGE client128a socket4086 630 255312-255375
START_RANGE client128a socket4087 620 255375-255437
START_RANGE client128a socket4088 630 255437-255500
START_RANGE client128a socket4089 620 255500-255562
START_RANGE client128a socket4090 630 255562-255625
START_RANGE client128a socket4091 620 255625-255687
START_RANGE client128a socket4092 630 255687-255750
START_RANGE client128a socket4093 620 255750-255812
START_RANGE client128a socket4094 630 255812-255875
START_RANGE client128a socket4095 620 255875-255937
START_RANGE client128a socket4096 630 255937-256000
#endif
/*-----*/
#define TES_FLAG_TRACE 0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x00000020
#define TES_FLAG_LOCAL_LOG 0x00000040
#define TES_FLAG_LOCAL_TRACE 0x00000080
#define TES_FLAG_LOCAL_IPRINT 0x00004000
#if 0
/* SETFLAG ALL TES_FLAG_TRACE */
SETFLAG ALL TES_FLAG_LOCAL_TRACE
SETFLAG ALL TES_FLAG_LOCAL_IPRINT
#endif
#if 0
SETFLAG client31 telnet 1 TES_FLAG_KEYSTROKE_TIME
#endif

```

D.2 RTE Scripts

tpccWeb.h

```

/*
*****
** Project : AIX DB/2 TPC-C
** Component : TPC-C/Client
** Name : tpccWeb.h
** Title : rte web defines
*****
** Copyright (c) IBM US - AUSTIN 2000
** Classification : IBM Internal Use Only
**
** History :
**          Develop by Austin RISC/6000 Performance Team
**

```

```

** Comments :
**
*****
*/
// Transaction Codes
// Transaction Codes
#define TXN_LOGIN 0
#define TXN_NEW_ORDER 1
#define TXN_PAYMENT 2
#define TXN_ORDER_STATUS 3
#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 TXN_NORD "nord"
#define TXN_PYMT "pymt"
#define TXN_ORDS "ords"
#define TXN_DLVY "dlvy"
#define TXN_STOK "stok"
#define TXN_sEXIT "exit"
#define TXN_MENU "menu"
#define ITEM_CMD_ID_START 11
#define ITEM_CMD_ID_END 55
#define APP_NAME "tpcc"
// Transaction Result Search Strings
// Transaction Result Search Strings
#define LOGIN_TITLE "Home Page"
#define MENU_TITLE "Main Menu"
#define NORD_TITLE "New Order"
#define PYMT_TITLE "Payment"
#define ORDS_TITLE "Order Status"
#define DLVY_TITLE "Delivery"
#define STOK_TITLE "Stock Level"
#define LOGIN_TITLE_LEN 9
#define MENU_TITLE_LEN 9
#define NORD_TITLE_LEN 9
#define PYMT_TITLE_LEN 7
#define ORDS_TITLE_LEN 12
#define DLVY_TITLE_LEN 8
#define STOK_TITLE_LEN 10
#define NORD_RESULTS_TITLE "New Order Results"
#define PYMT_RESULTS_TITLE "Payment Results"
#define ORDS_RESULTS_TITLE "Order Status Results"
#define DLVY_RESULTS_TITLE "Delivery Results"
#define STOK_RESULTS_TITLE "Stock Level Results"
#define NORD_RESULTS_TITLE_LEN 17
#define PYMT_RESULTS_TITLE_LEN 15

```

```

#define ORDS_RESULTS_TITLE_LEN
20
#define DLVY_RESULTS_TITLE_LEN
16
#define STOK_RESULTS_TITLE_LEN
19
#define CONTENT_LENGTH "Content-Length:
"
#define HEADER_TERMINATOR "\r\n\r\n"

////////////////////////////////////
// Field Lengths
////////////////////////////////////
#define HEADER_TERMINATOR_LENGTH
4
#define CONTENT_LENGTH_STR_LEN
16
////////////////////////////////////
// Transaction Request URLs
////////////////////////////////////
#define GET_REQUEST "GET %s
HTTP/1.1\r\nHost: %s\r\nConnection: Keep-Alive\r\nAccept:
text/\r\n\r\n"
#define GET_REQUEST_EXIT "GET %s
HTTP/1.1\r\nHost: %s\r\nConnection: close\r\nAccept:
text/\r\n\r\n"
#define LOGIN_URL "/tpcc/tpcc.html"
#define MENU_URL
"/tpcc/tpcc.html?00=menu&02=%d&03=%d"
#define NEW_ORDER_FORM_URL
"/tpcc/tpcc.html?00=nord&01=%d"
#define PAYMENT_FORM_URL
"/tpcc/tpcc.html?00=pymt&01=%d"
#define ORDER_STATUS_FORM_URL
"/tpcc/tpcc.html?00=ords&01=%d"
#define DELIVERY_FORM_URL
"/tpcc/tpcc.html?00=dlvy&01=%d"
#define STOCK_FORM_URL
"/tpcc/tpcc.html?00=stok&01=%d"
#define EXIT_FORM_URL
"/tpcc/tpcc.html?00=exit&01=%d"
#define NEW_ORDER_RESULTS_URL
"/tpcc/tpcc.html?00=nord&01=%d&03=%d&04=%d&05=%s"
#define PAYMENT_RESULTS_CID_URL
"/tpcc/tpcc.html?00=pymt&01=%d&03=%d&04=%d&05=%s&
06=%d&07=%d&08=%d.%02.2d"
#define PAYMENT_RESULTS_CLAST_URL
"/tpcc/tpcc.html?00=pymt&01=%d&03=%d&04=%d&05=%s&
06=%d&07=%d&08=%d.%02.2d"
#define ORDER_STATUS_RESULTS_CLAST_URL
"/tpcc/tpcc.html?00=ords&01=%d&03=%d&04=%d&05=%s"
#define ORDER_STATUS_RESULTS_CID_URL
"/tpcc/tpcc.html?00=ords&01=%d&03=%d&04=%d&05=%s"
#define DELIVERY_RESULTS_URL
"/tpcc/tpcc.html?00=dlvy&01=%d&10=%d"
#define STOCK_RESULTS_URL
"/tpcc/tpcc.html?00=stok&01=%d&09=%d"
#define NEW_ORDER_ITEM "%d=%d"
#define NEW_ORDER_ITEM_ENTRY
"%d=%d&%d=%d&%d=%d"
#define NEW_ORDER_EMPTY_ITEM "%d="

```

user_master.C

```

/*****
*****/
/* user_master.C Audit: 05/30/96 */
/*****
*****/
static char *rcsid="$Id: user_master.C,v 1.1 1999/02/22
06:31:05 channui Exp $";
#include <iostream.h>
#include <stdio.h>
#include <strings.h>
#include <stdlib.h>
#include <unistd.h>
#define _H_CUR01
#include <cur00.h>
#undef _H_CUR01
extern "C" {
#include "data/cur01.h"
int wrefresh (WINDOW *);
int wclrtoeol(WINDOW *);
int setupterm(char*,FILE*,int*);
int nodelay(int);
int keypad(int);
int wgetch(WINDOW *);
}
#include "data/rte.h"
#include "data/Stats.h"
#include "data/misc.h"
#include "user_tpcc.h"
struct header_s {
int slave;
int num;
int type;
int num_timestamps;
int user_data_length;
int data_type;
};
char *get_variable(char *name);
int get_variable(char *name, int *number);
int send_global_data(void);
int make_ratios (double *buffer);
extern int ramp_up_complete;
extern int interval_start_time, interval_stop_time;
//extern "C" int strcasecmp(char *s1, char *s2);
//extern "C" int strcmp(char *s1, char *s2, int n);
struct UserSpawnData {
int Warehouse;
int District;
};
/* user_master.C */
int user_statistics_print(void);
// int user_spawn(int *length, char *buffer);
int user_spawn(int min, int max, int number, int *length, char
*buffer);
int user_finished(int length, char *buffer);
extern SlaveStatus slave_status[MAX_SLAVES];
extern Stats status[MAX_TRAN_TYPE][MAX_TIMES];
extern WINDOW *statistics_win;
extern UserGlobal *shmglobal;
/* Transaction mix parameters */

```

```

double ratio_desired[6], ratio_min[6], ratio_max[6],
ratio_range[6];
char *ratio_names[] = { "RTE", "NEWORDER", "PAYMENT",
"ORDSTAT", "DELIVERY",
"STOCKLEV", NULL };
char *Status_Names[] = {"Menu", "Keying", "Response",
"Think"};

char *transaction_names[] = { "RTE", "New Order", "Payment",
"Order Stat",
"Delivery", "Stock Level", NULL };
static int current_status = 2, status_needs_refresh = 1;
int user_statistics_print(void) {
int i;
static int count = 0;
double ratios[6];
if (status_needs_refresh) {
count = 0;
status_needs_refresh = 0;
wmove (statistics_win, 0, 0);
wprintw (statistics_win, "%11s %8s %8s %8s %8s %8s
%6s %6s",
Status_Names[current_status], "90%", "Avg", "Min",
"Max",
"Samples", "Ratio", "Mix", "Think");
}
make_ratios(ratios);
for (i = 1; i <= 5; i++) {
if (count % 10 == 0) {
wmove (statistics_win, i, 0);
wprintw (statistics_win, "%11s %8.2f",
transaction_names[i],
status[i][current_status].ninety()/1000.0);
count = 0;
}
wmove (statistics_win, i, 21);
wprintw (statistics_win, "%8.2f %8.2f %8.2f %8d %6.2f
%6.2f %6.2f",
status[i][current_status].average()/1000.0,
status[i][current_status].min()/1000.0,
status[i][current_status].max()/1000.0,
status[i][current_status].samples(),
ratios[i], shmglobal->chances[i],
status[i][3].average()/1000.0);
}
wmove (statistics_win, 7, 0);
extern int runtime_counts[MAX_TRAN_TYPE];
extern int begin_time, ramp_up, run_time;
int start = interval_start_time;
int stop = interval_stop_time;
double interval = ((double)(stop-start) / (1000*60));
double samples = status[1][2].samples();
if (interval <= 0 || samples <= 0) {
wprintw (statistics_win, "TPM-C: %7s / ", "-----");
} else {
wprintw (statistics_win, "TPM-C: %7.2f / ",
samples/interval);
}
samples = runtime_counts[1];
if (samples > 0) {
start = begin_time+(ramp_up>=0)?ramp_up:0;
}

```

```

    if (run_time > 0 && stop > begin_time + ramp_up +
run_time) {
        stop = begin_time + ramp_up + run_time;
    }
    interval = (double)(stop - start)/(1000.0*60.0);
    wprintw (statistics_win, "%7.2f", samples/interval);
} else {
    wprintw (statistics_win, "-----");
}
}
count++;
return RTE_OK;
}
extern int login_begin;
int login_max_load;
#ifdef WHSEARRAYDBG
int outofboundwarn;
#endif
extern int min_warehouse;
extern int max_warehouse;
const int MAX_WAREHOUSES=100000;
/* All of this 10 stuff is district size. Should be a constant.
Maybe fix that later */
int num_warehouses = -1;
int warehouses[MAX_WAREHOUSES*10];
int user_spawn(int min, int max, int number, int *length, char
*buffer) {
//int user_spawn(int number, int *length, char *buffer) {
    int i, min_index;
    int adj_wh = num_warehouses; // adjusted warehouse
number
    UserSpawnData *ptr = (UserSpawnData *)buffer;
    *length = sizeof(*ptr);
// min_index = 0;
// for (i = 1; i < (num_warehouses)*10 && i <
MAX_WAREHOUSES*10; i++) {
//
// if both min and max are zero, running START, otherwise
running
// START_RANGE. Must also determine what the ending
warehouse number
// will be for said range
//
    if (min ==0 && max == 0) {
        min++;
        min_index = 0 ;
    } else {
        adj_wh = max; // inclusive range of wh-s
        min = min * 10;
        min_index = min;
    }
    for (i = min ; i < (adj_wh)*10 && i <
((MAX_WAREHOUSES+min_warehouse)*10); i++) {
        if (warehouses[i - (min_warehouse*10)] <
warehouses[min_index - (min_warehouse*10)]) {
            min_index = i;
        }
    }
    ptr->Warehouse = min_index / 10 + 1;
    ptr->District = min_index % 10 + 1;
#ifdef WHSEARRAYDBG

```

```

    if ((min_index - (min_warehouse*10) < 0) || (min_index -
(min_warehouse*10) >= (MAX_WAREHOUSES*10))) {
        if (outofboundwarn) {
            iprint (IPRINT_INFO, "(spawn) Out of range warehouse
number %d, (%d-%d (start) = %d (rel. num)\n",
                min_index, min_index, min_warehouse, min_index -
(min_warehouse*10));
            outofboundwarn=0;
        }
    }
}
#endif
warehouses[min_index - (min_warehouse*10)]++;
/* iprint (IPRINT_INFO, "Driver for Warehouse %d, District
%d started. warehouses[%d]++ = %d\n",
    ptr->Warehouse, ptr->District, min_index,
warehouses[min_index - (min_warehouse*10)]); */
return RTE_OK;
}
int user_finished(int length, char *buffer) {
    UserSpawnData *ptr = (UserSpawnData *)buffer;
    int temp = (ptr->Warehouse-1)*10+ptr->District-1;
#ifdef WHSEARRAYDBG
    if ((temp - min_warehouse*10 < 0) || (temp -
min_warehouse*10 >= MAX_WAREHOUSES*10)) {
        if (outofboundwarn) {
            iprint (IPRINT_INFO, "(finish) Out of range warehouse
number %d, (%d-%d (start) = %d (rel. num)\n",
                min_index, min_index, min_warehouse, min_index -
(min_warehouse*10));
            outofboundwarn=0;
        }
    }
}
#endif
warehouses[temp - (min_warehouse*10)]--;
/* iprint (IPRINT_INFO, "Driver for Warehouse %d, District
%d died. warehouses[%d]-- = %d\n",
    ptr->Warehouse, ptr->District, temp, warehouses[temp -
(min_warehouse*10)]); */
return RTE_OK;
}
double limit(double min, double max, double val) {
    if (val < min)
        return min;
    if (val > max)
        return max;
    return val;
}
int make_ratios (double *buffer) {
    int neword = status[NEWORDER][0].samples();
    int payment = status[PAYMENT][0].samples();
    int ordstat = status[ORDSTAT][0].samples();
    int delivery = status[DELIVERY][0].samples();
    int stocklev = status[STOCKLEV][0].samples();
    int total = neword + payment + ordstat + delivery + stocklev;
    int i;
    if (total == 0) {
        buffer[NEWORDER] = 100.0;
        for (i = 2; i < 6; i++) {
            buffer[i] = ratio_desired[i];
            buffer[NEWORDER] -= buffer[i];
        }
    }
    return 0;
}

```

```

    }
    buffer[PAYMENT] = (double)payment / (double)total * 100.0;
    buffer[ORDSTAT] = (double)ordstat / (double)total * 100.0;
    buffer[DELIVERY] = (double)delivery / (double)total * 100.0;
    buffer[STOCKLEV] = (double)stocklev / (double)total * 100.0;
    buffer[NEWORDER] = 100.0 - buffer[PAYMENT] -
buffer[ORDSTAT] -
        buffer[DELIVERY] - buffer[STOCKLEV];
    return total;
}
int user_global_update(int *length, char *buffer) {
    UserGlobal *shmglobal = (UserGlobal *)buffer;
    static double last[6];
    static last_test_state = 0;
    static int users_last=-1;
    double ratios[6];
    double current[6];
    int i, different = 0;
    int desired = 0;
    int host_busy, all_zero;
    *length = sizeof(*shmglobal);
    make_ratios(ratios);
    /* Calculate ratios we want for next time */
    if (ramp_up_complete) {
        current[NEWORDER] = 100.0;
        for (i = 2; i < 6; i++) {
            if (ratio_desired[i] > ratios[i]) {
                current[i] = ratio_max[i];
            } else {
                current[i] = 2*ratio_desired[i] - ratios[i];
                if (current[i] < ratio_min[i])
                    current[i] = ratio_min[i];
            }
            current[NEWORDER] -= current[i];
        }
    } else {
        for (i = 1; i < 6; i++) {
            current[i] = ratio_desired[i];
        }
    }
    /* Add up all the users */
    /* This needs to be changed to be more transparent */
    shmglobal->total_users = 0;
    for (i = 0; i < MAX_SLAVES; i++) {
        shmglobal->total_users += slave_status[i].active;
        desired += slave_status[i].desired;
    }
    /* Count up number of warehouses we WANT to have */
    if (num_warehouses < 0) {
        num_warehouses = (desired-1)/10+1;
    }
    shmglobal->max_warehouses = num_warehouses;
    host_busy = 0;
    all_zero = 1;
    for (i = 1; i <= 5; i++) {
        if (status[i][current_status].average() != 0) {
            all_zero = 0;
        }
        if ( status[i][current_status].average()/1000.0 >
login_max_load ) {
            host_busy = 1;
        }
    }
}

```

```

}
if (shmglobal->host_busy && all_zero) {
    host_busy = 1;
}
if (host_busy != shmglobal->host_busy) {
    shmglobal->host_busy = host_busy;
    different = 1;
}
for (i = 2; i < 6; i++) {
    if (current[i] != last[i])
        different = 1;
}
if (last_test_state != shmglobal->test_state) {
    different = 1;
    last_test_state = shmglobal->test_state;
}
// Don't send if it's the same as last time
if (!different && shmglobal->total_users == users_last) {
    return RTE_ERROR;
}
users_last = shmglobal->total_users;
for (i = 1; i < 6; i++) {
    shmglobal->chances[i] = last[i] = current[i];
}
return RTE_OK;
}
int user_isbusy() {
    return shmglobal->host_busy;
}

int parse_array(char *string, int max, int *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
        rc = sscanf(ptr, "%d", &buffer[i]);
        if (rc < 1) {
            free(temp);
            return i;
        }
        ptr = strtok(NULL, ",");
    }
    free(temp);
    return i;
}

int parse_array(char *string, int max, double *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
        rc = sscanf(ptr, "%lf", &buffer[i]);
        if (rc < 1) {
            free(temp);
            return i;
        }
        ptr = strtok(NULL, ",");
    }
    free(temp);
    return i;
}

```

```

int user_init() {
    double dbuffer[32];
    int rc, i;
    char *ptr;
    if (get_variable("KEYSTROKE_SLEEP", &shmglobal->keystroke_sleep) != RTE_OK) {
        shmglobal->keystroke_sleep = 0;
    }
    if (get_variable("LOGIN_TIMEOUT", &shmglobal->login_timeout) != RTE_OK) {
        shmglobal->login_timeout = 120; /* 2 minutes */
    }
    if (get_variable("KEYSTROKE_PACKET_SIZE", &shmglobal->keystroke_packet_size) != RTE_OK) {
        shmglobal->keystroke_packet_size = 0;
    }
    shmglobal->login_timeout *= 1000;
    if (get_variable("LOGIN_MAX_LOAD", &login_max_load) != RTE_OK) {
        login_max_load = 2;
    }
    if (get_variable("WAREHOUSES", &num_warehouses) != RTE_OK) {
        num_warehouses = -1;
    }
    if (get_variable("LASTC", &shmglobal->lastc) != RTE_OK) {
        shmglobal->lastc = 193; /* 2 minutes */
    }
    fprintf(IPRINT_INFO, "Login Timeout = %s\n",
            mstoaf_withfrac(shmglobal->login_timeout, 0));
    fprintf(IPRINT_INFO, "Keystroke Sleep = %s\n",
            mstoaf_withfrac(shmglobal->keystroke_sleep*1000, 0));
    fprintf(IPRINT_INFO, "Keystroke Packet Size= %d\n",
            shmglobal->keystroke_packet_size);
    if (num_warehouses >= 0) {
        fprintf(IPRINT_INFO, "Fixed Warehouses to = %d\n",
            num_warehouses);
    }
    if (!(ptr = get_variable("NEWORDER"))) {
        fprintf_error ("Error. NEWORDER variable not found\n");
        exit (1);
    }
    if (parse_array(ptr, 3, dbuffer)!=3) {
        fprintf_error ("Error. NEWORDER should be think,
            emulex_menu, emulex_response");
        exit (1);
    }
    shmglobal->think [NEWORDER] = dbuffer[0];
    shmglobal->emulex_menu [NEWORDER] = dbuffer[1];
    shmglobal->emulex_response[NEWORDER] = dbuffer[2];
    shmglobal->test_state = 0;
    for (i = 2; i < 6; i++) {
        if (!(ptr = get_variable(ratio_names[i])) ||
            (parse_array(ptr, 6, dbuffer)!=6)) {
            fprintf(__FILE__, __LINE__, IPRINT_ERROR,
                "Error. %s should be think, emulex_menu,
            emulex_response, desired, min, max",
                ratio_names[i]);
            exit (1);
        }
        shmglobal->think[i] = dbuffer[0];
    }
}

```

```

shmglobal->emulex_menu[i] = dbuffer[1];
shmglobal->emulex_response[i] = dbuffer[2];
ratio_desired[i] = dbuffer[3];
ratio_min[i] = dbuffer[4];
ratio_max[i] = dbuffer[5];
ratio_range[i] = ratio_max[i]-ratio_min[i];
}
for (i=0; i < (MAX_WAREHOUSES*10); i++) {
    warehouses[i] = 0;
}
#ifdef WHSEARRAYDBG
outofboundwarn=1;
#endif
return RTE_OK;
}
int user_extra_data(header_s *header) {
    int i;
    int num_timestamps;
    if (header->data_type != RTE_ITEM_KEYSTROKE_TIMES)
        return RTE_OK;
    int *times = (int *)((char *)header+sizeof(struct header_s));
    num_timestamps = header->user_data_length / 4 - 1;
    fprintf (IPRINT_TRACE, "Keystroke times = ");
    for (i = 0 ; i < num_timestamps; i++) {
        fprintf (IPRINT_TRACE, "%d ", times[i]);
    }
    fprintf (IPRINT_TRACE, "\n", times[i]);
}
return RTE_OK;
}
int user_process_command(char *command) {
    char buffer[256], *ptr;
    int i, found, len;
    strncpy (buffer, command, 256);
    ptr = strtok (buffer, "\t");
    found = 0;
    printf ("user_process_command('%s')\n", ptr);
    if (!strncasecmp (ptr, "pause")) {
        shmglobal->test_state = 1;
    } else if (!strncasecmp (ptr, "warmup")) {
        shmglobal->test_state = 2;
    } else if (!strncasecmp (ptr, "notest")) {
        shmglobal->test_state = 0;
    } else if (!strncasecmp (ptr, "login_max_load?")) {
        fprintf (IPRINT_WARNING, "Current LOGIN_MAX_LOAD
            = %d\n", login_max_load);
    } else if (!strncasecmp (command, "login_max_load=", 15)) {
        login_max_load=atoi(command+15);
        fprintf (IPRINT_WARNING, "Set LOGIN_MAX_LOAD =
            %d\n", login_max_load);
    } else if (!strncasecmp (ptr, "display")) {
        while (ptr && (ptr = strtok(NULL, "\t"))) {
            if (*ptr == '\0')
                continue;
            for (i = 0; i < 5; i++) {
                len = min(strlen(Status_Names[i]), strlen(ptr));
                if (!strncasecmp (ptr, Status_Names[i], len)) {
                    status_needs_refresh = found = 1;
                    current_status = i;
                    return RTE_OK;
                }
            }
        }
    }
}

```

```

        iprint (IPRINT_WARNING, "Unknown type to display:
%s\n", ptr);
    }
    } else {
        iprint (IPRINT_WARNING, "Unknown Command: '%s\n",
command);
        return RTE_ERROR;
    }
    return RTE_OK;
}
int transaction_process () {
    return RTE_OK;
}
int user_begin() {
    return RTE_OK;
}
/*
void user_make_header(char *buffer) {
    int i;
    struct user_data_header *data = (struct user_data_header
*)buffer;
}
*/

```

user_slave.C

```

/*****
*****
*****/
/* user_slave.C Audit: 05/30/96 */
/*****
*****/
static char *rcsid="$Id: user_slave.C,v 1.1 1999/02/22 06:31:06
channui Exp $";
/*****
*****/
/**
*** TPCC FILE FOR ALL USERS
***/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/time.h>
#include "rte_slave.h"
#include "user_tpcc.h"
/* This MUST match the corresponding one in client's inout.h
file! */
#define TRIGGER "\021"
#define NOSLEEP
// Increased EXPECT_TIMEOUT from 600000 - oz 10/20/97
#define EXPECT_TIMEOUT 6000000
#define KEYSWAIT_FUDGE 5000
extern SHM_Slave *shm;
extern TableEntrySlave *shmentry;
extern DriverStatus *status;
extern echo_trace(char *);
extern echo_trace();
extern char *expect_save;
extern char *expect_buffer_return();
const char *SQL_TPERRNO_MESSAGE = "tperno";

```

```

const char *SQL_RTN_MESSAGE = "rtn: ";
const char *SQL_FATAL_MESSAGE = "SQL Fatal Error";
const char *ROLLBACK_MESSAGE = "Item number
is not valid";
const char *CUSTOMER_ID_STRING = "Customer: ";
int WHSEID; /* warehouse number for each
users */
/*****
***/
/* The "uniform()" function has range of the absolute value of the
*/
/* difference between the min. and the max values upto
2147483647. */
/*****
***/
/*-----*/
/* NURand */
/*-----*/
/* A: 255 for C_LAST, 1023 for C_ID, 8191 for OL_I_ID */
/* x: 0 for C_LAST, 1 for C_ID and OL_I_ID */
/* y: 999 for C_LAST, 3000 for C_ID, 100000 for OL_I_ID */
/*-----*/
long
NURand(int A, int x, int y, long cval)
{
    return (((long) uniform((long) 0, (long) A) | (long)
uniform((long) x, (long) y) + cval) % (y - x + 1)) + x;
}
/*-----*/
/* getname */
/*-----*/
/* 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 */
/* three strings are concatenated to generate lastname */
/*-----*/
char *
getname()
{
    char *last_name_parts[] =
    {
        "BAR",
        "OUGHT",
        "ABLE",
        "PRI",
        "PRES",
        "ESE",
        "ANTI",
        "CALLY",
        "ATION",
        "EING"
    };
    static char lastname[128];
    int random_num;
    #if 1
        random_num = NURand(255, 0, 999, shmglobal->lastc);
    #else
        random_num = NURand(255, 0, 999, LASTC);
    #endif
    strcpy(lastname, last_name_parts[random_num / 100]);
    random_num %= 100;
    strcat(lastname, last_name_parts[random_num / 10]);
}

```

```

random_num %= 10;
strcat(lastname, last_name_parts[random_num]);
return (lastname);
}
typedef struct gen_tran_s {
    int invalid;
    void *data;
    long len;
    long keywait;
    long type;
    char *menu;
    char *results_request;
    char *form_request;
} gen_tran_t;
typedef struct gen_tran_url_s
{
    char *txn_form_url;
    char *txn_results_url;
} gen_tran_url_s;
int generic_transaction( gen_tran_t *data, char *host)
{
    char buffer[2048];
    static int consecutive_errs = 0;
    int rc;
    set_typing_delay(0);
    //iprint(IPRINT_TRACE, "> generic_transaction sleep (%d)
type(%d) *data (%d)\n", data->type, data->menu, data);
    #ifndef NOSLEEP
        if (shmglobal->test_state == 0)
            transaction_sleep_do();
    #endif
    #ifdef EXPECT_TIMEOUT
        int timeout = EXPECT_TIMEOUT;
    #else
        int timeout = 0;
    #endif
    // Start the transaction (MENU)
    //iprint(IPRINT_TRACE, "> generic_transaction start (%d)\n",
data->type);
    transaction_start(data->type, data->len, data->data);
    //send menu request page
    //iprint(IPRINT_TRACE, "> transmit data->menu: (%s)\n
request :(%s)", data->menu, data->form_request);
    //iprint(IPRINT_TRACE, "> transmit data->menu: (%s)\n
request :(%s)", data->menu);
    transmit(data->form_request);

    echo_trace ("Waiting for Menu");
    switch (*data->menu)
    {
        case '1':
            rc =
            expect_html(NORD_TITLE, timeout, NORD_TITLE_LEN);
            break;
        case '2':
            rc =
            expect_html(PYMT_TITLE, timeout, PYMT_TITLE_LEN);
            break;
        case '3':
            rc =
            expect_html(ORDS_TITLE, timeout, ORDS_TITLE_LEN);
            break;
    }
}

```



```

        case '4':
            rc =
expect_html(DLVY_TITLE,timeout,DLVY_TITLE_LEN);
            break;
        case '5':
            rc =
expect_html(STOK_TITLE,timeout,STOK_TITLE_LEN);
            break;
        default:
            rc = ERROR;
    }
    if(rc == ERROR)
    {
        iprint (IPRINT_ERROR, "Slave %d:Failed to receive
%s input form\n**Request:-->%s<--\n",
shmentry->num, data->menu,data-
>form_request);
        return (ERROR);
    }
//if (expect_html(TRIGGER, timeout) == ERROR)
//{
//    iprint (IPRINT_ERROR, "Slave %d: Failed to receive %s
screen\n",
//    shmentry->num, data->menu);
//    return (ERROR);
//}

#ifdef NOSLEEP
    usleep(shmglobal->emulex_menu[data-
>type]*1000000.0+0.9);
#endif
    // Send our request (KEYING)
    transaction_mark(WHERE_NOW);
    echo_trace ("Keying");
#ifdef NOSLEEP
    usleep(data->keywait*1000000+KEYWAIT_FUDGE); //
Keying delay
#endif
    // Wait for response (RESPONSE)
    transaction_mark(WHERE_NOW);
    //iprint(IPRINT_TRACE, "> transmit request :(%s)\n",data-
>results_request);
    transmit(data->results_request);
    echo_trace ("Wait for Response");
    switch (*data->menu)
    {
        case '1':
            rc =
expect_html(NORD_RESULTS_TITLE,timeout,NORD_RESULT
S_TITLE_LEN);
            break;
        case '2':
            rc =
expect_html(PYMT_RESULTS_TITLE,timeout,PYMT_RESULT
S_TITLE_LEN);
            break;
        case '3':
            rc =
expect_html(ORDS_RESULTS_TITLE,timeout,ORDS_RESULT
S_TITLE_LEN);
            break;
        case '4':

```

```

            rc =
expect_html(DLVY_RESULTS_TITLE,timeout,DLVY_RESULTS
_TITLE_LEN);
            break;
        case '5':
            rc =
expect_html(STOK_RESULTS_TITLE,timeout,STOK_RESULT
S_TITLE_LEN);
            break;
        default:
            rc = ERROR;
    }
    if(rc == ERROR)
    {
        iprint (IPRINT_ERROR, "Slave %d:Failed to receive
%s result page\n**Request:-->%s<--\n",
shmentry->num, data->menu,data-
>form_request);
        return (ERROR);
    }

// if (expect_html(TRIGGER, timeout) == ERROR) {
//iprint (IPRINT_ERROR, "Slave %d: Failed to receive %s
response\n",
//    shmentry->num, data->menu);
//    return (ERROR);
//}

#ifdef NOSLEEP
    usleep(shmglobal->emulex_response[data-
>type]*1000000.0+0.9);
#endif
    // Look for errors and set our think time (THINK)
    transaction_mark(WHERE_NOW);
    if (expect_buffer_search("ERROR:",6))
    {
        FILE *fd;
        if ((fd = fopen("/u/rte/error.txt", "a"))!=NULL)
        {
            fprintf(fd,"Slave %d: %s found %s\n%s\n",
shmentry->num, data->menu,
"ERROR:",expect_buffer_return());
            fclose(fd);
        }
        data->invalid = 1;
        iprint (IPRINT_ERROR, "Slave %d: %s found
%s\n%s\n",
shmentry->num, data->menu,
"ERROR:",expect_buffer_return());
        // Very dangerous, keep going rather than exiting...
        //return RTE_ERROR;
        // Check for consecutive errors and if there are more
        than
        // 4 of them exit - allow for transient errors to make
        // tuning and testing easier -oz
        // In either case the transaction is marked as invalid and
        // will be reported as an error by the analyze program.
        if (consecutive_errs++ > 4)
            return RTE_ERROR;
    }
    else
    {

```

```

        consecutive_errs = 0;
    }
    echo_trace ("Thinking");
    transaction_sleep_set(neg_exp_4(shmglobal->think[data-
>type])*1000.0);
    //iprint(IPRINT_TRACE, "< generic_transaction finish\n");
    return (RTE_OK);
}
/*****
**/
/**      Delivery Transaction      ***/
/*****
**/
int
Delivery(char *host,int terminal)
{
    static struct delivery_struct delivery, delivery_new;
    int rc;
    char *ptr;
    gen_tran_t tran;
    tran.invalid = 0;
    tran.data = &delivery;
    tran.len = sizeof(delivery);
    tran.keywait = 2;
    tran.type = DELIVERY;
    tran.menu = "4";
    char dlvy_url[128];
    char form_buffer[256];
    char results_buffer[256];
    tran.form_request = form_buffer;
    tran.results_request = results_buffer;
    //create dlvy form request
    sprintf(dlvy_url,DELIVERY_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,dlvy_url,host);
    // Set up all data for new transactions
    delivery_new.carrier = uniform(1, 10); // carrier # 1 to 10
    //create dlvy results request
    sprintf(dlvy_url,DELIVERY_RESULTS_URL,terminal,deliv
ery_new.carrier);
    sprintf(results_buffer,GET_REQUEST,dlvy_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    delivery = delivery_new;
    delivery.invalid = tran.invalid;
    //iprint(IPRINT_TRACE,"dlvy txn finished, rc:%d
tran.invalid:%d\n",rc,delivery.invalid);
    return (rc);
}
/*****
**/
/**      New Order Transaction      ***/
/*****
**/
int NewOrder(char *host,int terminal)
{
    static struct neword_struct neword, neword_new;
    int i, rc, whses, low_whse=1;
    char nord_form_url[128];
    char form_buffer[512];
    char nord_results_url[2048];
    char results_buffer[4096];

```

```

char *ptr;
char *ptr2;
const char *err_ptr;
gen_tran_t tran;
tran.invalid = 0;
tran.data = &neword;
tran.len = sizeof(neword);
tran.keywait = 18;
tran.type = NEWORDER;
tran.menu = "1";

tran.form_request = form_buffer;
tran.results_request = results_buffer;

neword_new.rollback=0;
/** SECTION TO DETERMINE ROLLBACK TRANSACTION
FOR 1% OF NEW ORDERS ***/
neword_new.did = uniform(1, 10);
// district number
neword_new.cid = NURand(1023, 1, 3000,
CUSTC); // customer # 1 to 3000
neword_new.nloop = uniform(5, 15);
neword_new.olremote = 0;
whses = shmglobal->max_warehouses;
for (i = 0; i < neword_new.nloop; i++)
{
// Warehouse Number
neword_new.item[i].olswid = WHSEID;
if (whses > 1 && (uniform(0.0, 100.0) < 1.0))
{
/* for 1% of items (if * uniform()==0) */
/* Generate a uniform whse number that's different
from WHSEID */
neword_new.item[i].olswid =
(long) uniform((long) low_whse, (long)whses-
1);
if (neword_new.item[i].olswid >= WHSEID)
neword_new.item[i].olswid++;
neword_new.olremote++; // find total number of
remote order-lines
}
// Item number 1-100000
neword_new.item[i].oliid = NURand(8191, 1,
100000, ITEMC);
// Quantity 1-10
neword_new.item[i].olquantity = uniform(1, 10);
}
/* end of for n_loop */
// We occasionally force a transaction to have invalid data to
force a
// rollback
if (uniform(1, 5000) <= 50)
neword_new.item[neword_new.nloop-1].oliid =
999999;
neword_new.oremote = (neword_new.olremote > 0);
//create new order form request
sprintf(nord_form_url,NEW_ORDER_FORM_URL,termina
l);

//create get form request
sprintf(form_buffer,GET_REQUEST,nord_form_url,host);
//create new order results url
char itemString[1024];

```

```

ptr2=itemString;
short item_cmd_start = ITEM_CMD_ID_START;
for (i = 0; i < neword_new.nloop; i++)
{
ptr2 += sprintf(ptr2, NEW_ORDER_ITEM,
item_cmd_start++,
neword_new.item[i].olswid);

ptr2 += sprintf(ptr2,NEW_ORDER_ITEM,
item_cmd_start++,
neword_new.item[i].oliid);

ptr2 += sprintf(ptr2, NEW_ORDER_ITEM,
item_cmd_start++,
neword_new.item[i].olquantity);
}
//seal up url w/ empty items
for (i = item_cmd_start; i <= ITEM_CMD_ID_END; i++)
{
ptr2 += sprintf(ptr2,NEW_ORDER_EMPTY_ITEM,i);
}
// number of items to order (5-15)
//find total number of items to order
sprintf(nord_results_url,NEW_ORDER_RESULTS_URL,te
rminal,
neword_new.did,neword_new.cid,
itemString);

//create get results request
sprintf(results_buffer,GET_REQUEST,nord_results_url,ho
st);

// Go do the transaction
rc = generic_transaction(&tran,host);
neword = neword_new;
neword.invalid = tran.invalid;
// Check for a rollback
if ((err_ptr = expect_buffer_search("Item number is not
valid",24)))
{
neword.rollback=1;
echo_trace ("Found rollback!\n");
}
// Grab the orderID from the
if (!(err_ptr = expect_buffer_search("Order Number: ",14)))
{
echo_trace ("Didn't find order-id for neworder");
iprint (IPRINT_ERROR, "Neworder didn't have
Order-ID\n%s\n",expect_buffer_return());
//iprint (IPRINT_ERROR, "Neworder didn't have
Order-ID\n");
neword.oid = -1;
}
else
{
neword.oid = atoi(err_ptr+14);
// iprint(IPRINT_ERROR,"New order order
id:%d\n",neword.oid);
}

```

```

// This is really not useful since we aren't going to be sending
individual
// keystrokes anymore
if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME) {
log_data(RTE_ITEM_KEYSTROKE_TIMES,
keystroke_length*sizeof(int),keystroke_times);
}
//iprint(IPRINT_TRACE,"nord txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
return (rc);
}

/*****
**/
/** Order Status Transaction **/
/*****
**/
int OrderStatus(char *host,int terminal) {
static struct ordstat_struct ordstat, ordstat_new;
//char buffer[2048];
int rc;
char *ptr;
gen_tran_t tran;
tran.invalid = 0;
tran.data = &ordstat;
tran.len = sizeof(ordstat);
tran.keywait = 2;
tran.type = ORDSTAT;
tran.menu = "3";
//tran.request = buffer;

//Joe N.
char ords_url[256];
char form_buffer[512];
char results_buffer[2048];
tran.results_request = results_buffer;
tran.form_request = form_buffer;

//create order status form request
sprintf(ords_url,ORDER_STATUS_FORM_URL,terminal);
sprintf(form_buffer,GET_REQUEST,ords_url,host);

// Set up all data for new transactions
ordstat_new.did = uniform(1, 10); /* district number 1 to
10 */
if (uniform(1, 100) <= 60)
{
/* for 60% of transactions */
char *tmp = getname();
strcpy(ordstat_new.clast, tmp); /* by customer
last name */
if (ordstat_new.clast[0] < 'A' || ordstat_new.clast[0] >
'Z')
{
iprint (IPRINT_ERROR,
"ASSERTION: OrderStatus getname() returns invalid
name! '%s\n",
ordstat_new.clast);
return RTE_ERROR;
}
ordstat_new.byname = 1;
ordstat_new.cid = 0;
}

```

```

}
else
{
    ordstat_new.cid = NURand(1023, 1, 3000, CUSTC);
    /* cust. # 1 to 3000 */
    ordstat_new.byname = 0;
    ordstat_new.clast[0] = (char) NULL;
}
//iprint(IPRINT_TRACE,"Order status fields,w_id:%d
d_id:%d n_d_id:%d c_id:%d\n", data->Warehouse,data-
>District,ordstat_new.did,ordstat_new.cid);
//create order status url request
if (ordstat_new.byname)

    sprintf(ords_url,ORDER_STATUS_RESULTS_CLAST_U
RL,terminal,ordstat_new.did,

else

    sprintf(ords_url,ORDER_STATUS_RESULTS_CID_URL,t
erminal,ordstat_new.did,

    sprintf(results_buffer,GET_REQUEST,ords_url,host);

// Go do the transaction
rc = generic_transaction(&tran,host);
ordstat = ordstat_new;
ordstat.invalid = tran.invalid;
//iprint(IPRINT_TRACE,"ords txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
return (rc);
}
/*****
**/
/****      Payment Transaction      ****/
/*****
**/
int
Payment(char *host,int terminal)
{
    static struct payment_struct payment, payment_new;
    int    dollars, cents, rc, whses, low_whse = 1;
    char   *ptr;
    gen_tran_t   tran;
    tran.invalid = 0;
    tran.data = &payment;
    tran.len = sizeof(payment);
    tran.keywait = 3;
    tran.type = PAYMENT;
    tran.menu = "2";

    char pymt_url[128];
    char form_buffer[256];
    char results_buffer[2048];
    tran.results_request = results_buffer;
    tran.form_request = form_buffer;
    //create pymt form url
    sprintf(pymt_url,PAYMENT_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,pymt_url,host);

    payment_new.did = uniform(1, 10); /* district number 1 to
10 */

```

```

if (uniform(1, 100) <= 60) /* for 60% of transactions */
    strncpy(payment_new.clast, getname(), 17); // by
customer last name
if (payment_new.clast[0] < 'A' || payment_new.clast[0] >
'Z'){
    iprint (IPRINT_ERROR,
"ASSERTION: payment_new getname() returns
invalid name! '%s'\n",
    payment_new.clast);
    return RTE_ERROR;
}
    payment_new.byname = 1;
    payment_new.cid = 0;
} else {
    payment_new.cid = NURand(1023, 1, 3000, CUSTC);
    /* cust. # 1 to 3000 */
    ordstat_new.byname = 0;
    payment_new.clast[0] = (char) NULL;
}
whses = shmglobal->max_warehouses;
if (whses < 2 || uniform(1, 100) <= 85) /* for 85 % of
transactions */
    payment_new.cwid = WHSEID;
    payment_new.cdoid = payment_new.did;
    payment_new.remote = 0;
} else { /* for 15 % of transactions */
    payment_new.cwid = (long) uniform((long)low_whse,
(long) whses-1);
    if (payment_new.cwid >= WHSEID)
        payment_new.cwid++;
    payment_new.remote = 1;
    payment_new.cdoid = uniform(1, 10); /* district 1 to 10
*/
}
dollars = uniform(1, 5000);/* dollar amt = 1 to 5000 */
if (dollars == 5000)
    cents = 0;
else
    cents = uniform(0, 99);
    payment_new.amount = ((double) dollars) + ((double) cents) /
100.0;

//create payment results url
if (payment_new.byname)

    sprintf(pymt_url,PAYMENT_RESULTS_CLAST_URL,term
inal,

    payment_new.did,payment_new.clast,payment_new.cwid,

    payment_new.cdoid,dollars,cents);
else

    sprintf(pymt_url,PAYMENT_RESULTS_CID_URL,terminal
,

    payment_new.did,payment_new.cid,payment_new.cwid,

    payment_new.cdoid,dollars,cents);
    sprintf(results_buffer,GET_REQUEST,pymt_url,host);

// Go do the transaction

```

```

rc = generic_transaction(&tran,host);
payment = payment_new;
payment.invalid = tran.invalid;
//iprint(IPRINT_TRACE,"pymt txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);

return (rc);
}
/*****
**/
/****      Stock Level Transaction      ****/
/*****
**/
int
StockLevel(char *host,int terminal)
{
    static struct stocklev_struct stocklevel, stocklevel_new;
    int    rc;
    char   *ptr;
    gen_tran_t   tran;
    tran.invalid = 0;
    tran.data = &stocklevel;
    tran.len = sizeof(stocklevel);
    tran.keywait = 2;
    tran.type = STOCKLEV;
    tran.menu = "5";
    char stok_url[128];
    char form_buffer[256];
    char results_buffer[2048];
    tran.results_request = results_buffer;
    tran.form_request = form_buffer;
    //create stok form url
    sprintf(stok_url,STOCK_FORM_URL,terminal);
    sprintf(form_buffer,GET_REQUEST,stok_url,host);

    stocklevel_new.invalid = 0;
    stocklevel_new.threshold = uniform(10, 20); /* uniform no.
between 10 and

        * 20 */

    //create stok results url
    sprintf(stok_url,STOCK_RESULTS_URL,terminal,stocklev
el_new.threshold);
    sprintf(results_buffer,GET_REQUEST,stok_url,host);

// Go do the transaction
rc = generic_transaction(&tran,host);
stocklevel = stocklevel_new;
stocklevel.invalid = tran.invalid;
//iprint(IPRINT_TRACE,"stok txn finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
return (rc);
}
/*****
**/
/****      MAIN()      ****/
/*****
**/
int
user_transaction(char *host,void *data,int terminal)
{
    UserLocal *localdata = (UserLocal *)data;
    char    logout[32];

```

```

double ntask;
int resp;
static int task = 0;
if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME)
{
    int rc;
    /* Wait for specified period of time */
    sleep (shmglobal->keystroke_sleep);
    /* Quit after one transaction */
    shm->lock(shmentry->pid);
    shmentry->flags |= TES_FLAG_DIE;

    shm->unlock(shmentry->pid);
    rc = NewOrder(host,terminal);
    iprint (IPRINT_INFO, "Slave %d: Keystroke timing
setting die flag\n", shmentry->num);
    return rc;
}

#if 1
switch (shmglobal->test_state)
{
case 0: // Normal
    break;

case 1: // pause
    sleep (1);
    return RTE_OK;
case 2: // warmup
    switch(task++)
    {
        case 0: return Delivery(host,terminal);
        case 1: return OrderStatus(host,terminal);
        case 2: return Payment(host,terminal);
        case 3: return StockLevel(host,terminal);
        case 4: task = 0; return
NewOrder(host,terminal);
        return NewOrder(host,terminal);
    }
}

/**
*** CHOOSE ONE OF THE TRANSACTIONS
***/

ntask = (double) uniform(0.0, 100.0);
if (ntask <= shmglobal->chances[DELIVERY])
{
    return Delivery(host,terminal);
    //return NewOrder(host,terminal);
}
ntask -= shmglobal->chances[DELIVERY];
if (ntask <= shmglobal->chances[ORDSTAT])
{
    return OrderStatus(host,terminal);
}
ntask -= shmglobal->chances[ORDSTAT];
if (ntask <= shmglobal->chances[PAYMENT])
{

```

```

        return Payment(host,terminal);
    }
    ntask -= shmglobal->chances[PAYMENT];
    if (ntask <= shmglobal->chances[STOCKLEV])
    {
        return StockLevel(host,terminal);
    }
    return NewOrder(host,terminal);
#else
{
    int deck[100], count=-1, i, size=1, tmp;
    // lock deck
    if (count < 0) {
        // deck is empty fill it up
        count = 0;
        for (i = 0; i < 43 * size; i++) {
            deck[count++] = Payment;
        }
        for (i = 0; i < 4 * size; i++) {
            deck[count++] = StockLevel;
        }
        for (i = 0; i < 4 * size; i++) {
            deck[count++] = OrderStatus;
        }
        for (i = 0; i < 4 * size; i++) {
            deck[count++] = Delivery;
        }
        for (; count < 100 * size; i++) {
            deck[count++] = NewOrder;
        }
        // randomize the deck
        for (i = 0; i < 100 * size; i++) {
            int tmp;
            int pick = uniform(i+1, 100);
            tmp = deck[i];
            deck[i] = deck[pick];
            deck[pick] = tmp;
        }
        tmp = deck[count--];
        // unlock deck
        switch(tmp) {
            case Delivery: return Delivery(host,terminal);
            case OrderStatus: return OrderStatus(host,terminal);
            case Payment: return Payment(host,terminal);
            case StockLevel: return StockLevel(host,terminal);
            case NewOrder: return NewOrder(host,terminal);
        }
    }
}
#endif
#if 0
if (resp != RTE_OK) { /* logoff if response is not
correct */

```

```

        strcpy(logout, "9\n"); /* menu option 9 */
        transmit(logout);
        resp = expect("tpcc_cstux_inf:");
        return (ERROR);
    } else
        return (RTE_OK);
#endif
} /* end of Main */
int user_parameter_change(void) {
    #if 0
    int i;
    iprint(IPRINT_TRACE, "Slave %d: total_users = %d\n",
shmentry->num);
    iprint(IPRINT_TRACE, "Slave %d: chances = ", shmentry-
>num);
    for (i = 0; i < MAX_TRAN_TYPE; i++)
        iprint(IPRINT_TRACE, "%6.2f ", shmglobal->chances[i]);
    iprint(IPRINT_TRACE, "\nSlave %d: think = ", shmentry-
>num);
    for (i = 0; i < MAX_TRAN_TYPE; i++)
        iprint(IPRINT_TRACE, "%6.2f ", shmglobal->think[i]);
    iprint(IPRINT_TRACE, "\n");
    #endif
    return RTE_OK;
}
int user_login(char *user, char *password, void *data) {
    UserLocal *localdata = (UserLocal *)data;
    int rc;
    int timeout_value = shmglobal->login_timeout;
    char buffer[32];
    set_typing_delay(0);
    rc = expect (TRIGGER, timeout_value);
    if (rc == RTE_ERROR) {
        iprint (IPRINT_ERROR, "Slave %d: didn't find Warehouse
prompt\n", shmentry->num);
    }
    sprintf(buffer, "%d\t%d\n", localdata->Warehouse, localdata-
>District);
    transmit(buffer);
    iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d,
District=%d, pid=%d\n", shmentry->num, localdata-
>Warehouse, localdata->District, getpid());
    rc = expect (TRIGGER, timeout_value);
    if (rc != RTE_OK) {
        iprint (IPRINT_ERROR, "Slave %d: Failed logging in\n",
shmentry->num);

        return RTE_ERROR;
    }
    return RTE_OK;
}
int user_login_html(char *host,void *data,int *terminal)
{
    UserLocal *localdata = (UserLocal *)data;
    int rc;
    int timeout_value = shmglobal->login_timeout;
    char request[256];
    char url[30];
    set_typing_delay(0);
    iprint(IPRINT_ERROR,"Generating login request for
host:%s\n",host);
    //generate login page request

```

```

    sprintf(request,GET_REQUEST,LOGIN_URL,host);
    iprint(IPRINT_ERROR,"sending login form
request:%s\n",request);
    //send the request
    transmit(request);
    iprint(IPRINT_ERROR,"login request sent, reading response in
expect_html()\n");
    //read the request
    rc =
expect_html(LOGIN_TITLE,timeout_value,LOGIN_TITLE_LEN);
    if (rc != RTE_OK)
    {
        iprint(IPRINT_ERROR,"Login request failed, unable
to find login key words:%s\n",LOGIN_TITLE);
        return RTE_ERROR;
    }

    iprint(IPRINT_ERROR,"login request read\n");
    //generate url and page get request
    sprintf(url,MENU_URL,localdata->Warehouse, localdata-
>District);

    sprintf(request,GET_REQUEST,url,host);
    iprint(IPRINT_ERROR,"sending login results
request:%s\n",request);
    transmit(request);
    iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d,
District=%d, pid=%d\n", shmentry->num, localdata-
>Warehouse, localdata->District, getpid());
    rc = expect_html(MENU_TITLE,
timeout_value,MENU_TITLE_LEN);
    if (rc != RTE_OK)
    {
        iprint (IPRINT_ERROR, "Slave %d: Failed logging in\n",
shmentry->num);
        return RTE_ERROR;
    }
    iprint (IPRINT_TRACE, "User login successful Slave%d:
Warehouse=%d, District=%d, pid=%d\n",shmentry->num,
localdata->Warehouse, localdata->District, getpid());
    rc = get_term_id(terminal);
    if(rc != RTE_OK)
        return RTE_ERROR;
    iprint(IPRINT_TRACE, "Terimnal set for this user:%d w/
warehouse:%d district:%d\n",terminal,localdata-
>Warehouse,localdata->District);

    return RTE_OK;
}
int get_term_id(int *terminal)
{
    //search for terminal id
    const char *termID_ptr;
    if (!(termID_ptr = expect_buffer_search("NAME=\01\
VALUE=\0",17)))
    {
        echo_trace ("Did not find terminal id in
response...");
        iprint (IPRINT_ERROR, "No terminal id specified.");
        return RTE_ERROR;
    }
}
else

```

```

{
    *terminal = atoi(termID_ptr+17);
    iprint(IPRINT_ERROR,"Terminal id:%d\n",terminal);
}
return RTE_OK;
}
int user_init () {
    extern int expect_save_active;
    WHSEID = shmlocal->Warehouse;
    status->max_transmit = shmglobal->keystroke_packet_size;
    expect_save_active = 1;
    return RTE_OK;
}
int user_logout () {
    iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d,
District=%d logging out\n", shmentry->num, shmlocal-
>Warehouse, shmlocal->District);
    return RTE_OK;
}
int user_cleanup () {
    transaction_sleep_do();
    transaction_start(0, 0, NULL); // Just something to clear out
the buffer...
    return RTE_OK;
}
int user_spawn_ok() {
    int rc, hb;
    hb = ((UserGlobal *) (shm->global_data))->host_busy;
    rc = hb?RTE_ERROR:RTE_OK;
    return rc;
}

```

user tpcc.h

```

/*****
*****
*****/
/* user_tpcc.h Audit: 05/30/96 */
/*****
*****/
/* $Id: user_tpcc.h,v 1.1 1999/02/22 06:31:06 channui Exp $ */
#ifndef USER_TPCC_H
#define USER_TPCC_H
/*****
*****/
/**** run-time constant for customer last name from 0 to 255,
****/
/**** run-time constant for customer id from 0 to 1023, ****/
/**** run-time constant for item id from 0 to 8191. ****/
/*****
*****/
/* #define LASTC 117 */
/* Change for 3.1 */
#define LASTC 193
#define CUSTC 319
#define ITEM 3849
/*****
*****/
/**** response type ****/

```

```

/*****
*****/
/* #define OK 1 */
/* #define ERROR -1 */
/*****
*****/
/**** transaction type ****/
/*****
*****/
#define NEWORDER 1
#define PAYMENT 2
#define ORDSTAT 3
#define DELIVERY 4
#define STOCKLEV 5
/*****
*****/
/**** transaction structures ****/
/*****
*****/
struct neword_struct {
    char invalid; /* transaction completed successfully */
    long did;
    long cid;
    long oid; /* Order-ID returned from client */
    long nloop; /* number of order line, avg = 15 */
    char oremote; /* 1 for remote order, 10% */
    long olremote; /* number of remote order line, 1% */
    char rollback; /* actually saw rollback text on screen */
    struct items_struct {
        long olswid;
        long oliid;
        long olquantity;
    } item[15];
};
struct payment_struct {
    char invalid; /* transaction completed successfully */
    long did;
    long cid;
    long cwid;
    long coid;
    char clast[17];
    double amount;
    char byname; /* 1 for by last name, 0 for by id */
    char remote; /* 1 for remote warehouse, 0
otherwise */
};
struct ordstat_struct {
    char invalid; /* transaction completed successfully */
    long did;
    long cid;
    char clast[17];
    char byname; /* 1 for by last name, 0 for by id */
};
struct delivery_struct {
    char invalid; /* transaction completed successfully */
    char carrier;
};
struct stocklev_struct {
    char invalid; /* transaction completed successfully */
    long threshold;
};
struct generic_struct {

```

```

char    invalid; /* transaction completed sucessfully */
};
union transaction_info {
char    invalid;
struct generic_struct generic;
struct neword_struct neword;
struct payment_struct payment;
struct ordstat_struct ordstat;
struct delivery_struct delivery;
struct stocklev_struct stocklev;
};
struct UserGlobal {
int total_users;
int max_warehouses;
int keystroke_sleep;
int login_timeout;
int keystroke_packet_size;
int lastc;
int test_state;
int host_busy;
double chances[MAX_TRAN_TYPE];
double think[MAX_TRAN_TYPE];
double emulex_response[MAX_TRAN_TYPE];
double emulex_menu  [MAX_TRAN_TYPE];
};
struct UserLocal {
int Warehouse;
int District;
};
struct user_data_header {
};
extern UserGlobal *shmglobal;
extern UserLocal *shmlocal;
#endif

```

Appendix - E: Third Party Pricing Information



800.750.4239

SHOPPING CART

[▶ Your Saved Carts](#) [▶ Save This Cart](#) [▶ Edit Saved Carts](#) [▶ Send To An Associate](#)

[Continue to Checkout](#)

Quantity	Product	CDW	Usually Ships	Price	Ext. Price
<input type="text" value="10"/>	Cisco Catalyst 2970 24 10/100/1000 BASE-T ports	511987	Same Day	\$2,809.38	\$28,093.80
Click to remove an item from your cart				Sub-Total	\$28,093.80

Update

Clear Cart

[Continue to Checkout](#)

[Continue Shopping](#) | [Go to CDW.com Homepage](#)

Related Top Sellers For: **Cisco Catalyst 2970 24 10/100/1000 BASE-T ports**

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

November 15, 2004

IBM Corporation
Tony Petrossian
11501 BURNET ROAD
Austin, TX 78758

Mr. Petrossian:

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
C11-00821	Windows 2000 Server <i>Server License Only - No CALs</i> <i>Discount Schedule: No Level</i> <i>Unit Price reflects a 8% discount from the retail unit price of \$799.</i>	\$738	128	\$94,464
254-00170	Visual C++ Standard Edition <i>Discount Schedule: No Discounts Applied</i>	\$109	1	\$109
	Microsoft Problem Resolution Services <i>Professional Support</i> <i>(1 incident)</i>	\$245	1	\$245

All products are currently orderable through Microsoft's normal distribution channels.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCtope0415111375

Please include this Reference ID in any correspondence regarding this price quote.