

IBM eServer pSeries 690
Model 7040-681
Using
AIX 5L V5.2
and
DB2 Universal Database 8.1

TPC BenchmarkTM C
Full Disclosure Report



First Edition
February 17, 2004

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 Trademark of the Transaction Processing Performance Council

Microsoft Windows 2000 server and COM+ Registered Trademark of Microsoft Corporation

First Edition: February 17, 2004

Second Edition: May 20, 2004

Third Edition: July 12, 2004

Forth Edition: March 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:

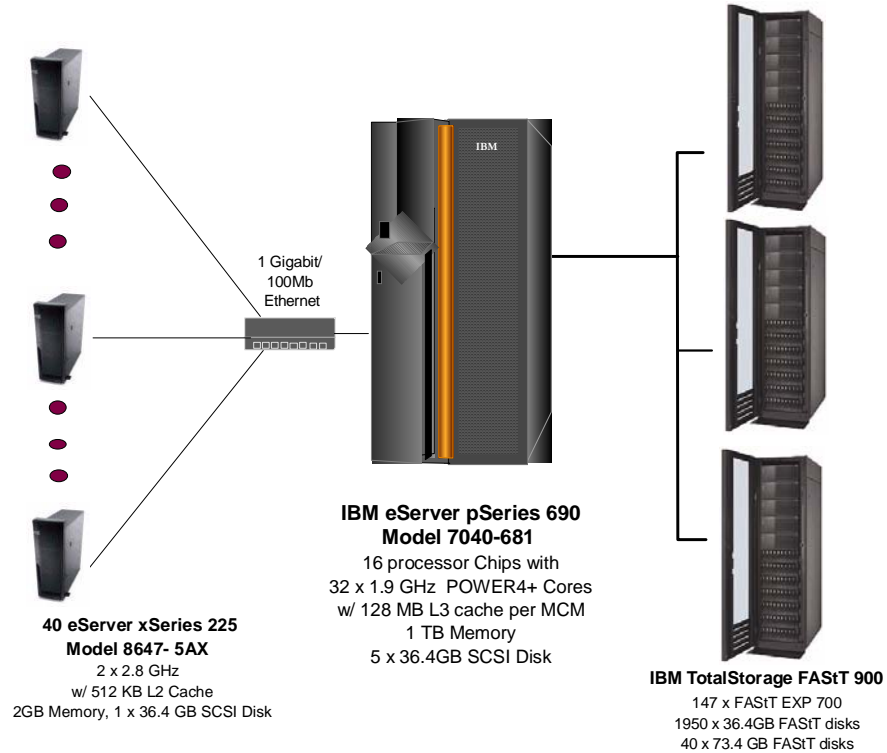
TPC Benchmark Administrator
IBM Commercial Performance
Mail Stop 9571
11501 Burnet Road
Austin, TX 78758
FAX Number (512) 838-1852

© 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 pSeries 690 Model 7040-681		TPC-C Rev. 5.2	
			Report Date: February 17, 2004 Updated: July 12, 2004	
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
\$5,565,772	1,025,486.07 tpmC	\$5.43/tpmC	August 16, 2004	
Processors Chip/Core/Thread	Database Manager	Operating System	Other Software	No. Users
16/32/32	DB2 UDB 8.1	AIX 5L V5.2	Microsoft COM+	817,000



System Components	Each Client		Server	
	Quantity	Description	Quantity	Description
Processors chip/Core/Thread	2/2/2	2.8 GHz Xeon W/512KB L2 Cache	16/32/32	1.9 GHz POWER4+ w/ 128 MB L3 cache per MCM, 4 MCMs
Memory	4	512 MB	8	128 GB
Disk Controllers	1	Ultra3 SCSI-2 (Integrated)	1 14 28	SCSI-2 (Integrated) FAStT900 Fiber Channel Adapters
Disk Drives	1	36.4 GB Ultra SCSI	5 1950 40	36.4 GB SCSI 10K RPM disk 36.4 GB FAStT disk 15K RPM 73.4 GB FAStT disks 15K RPM
Total Storage		36.4 GB		74,098GB
Terminals	1	System Console	1	System Console



IBM eServer pSeries 690 Model 7040-681

TPC-C Rev. 5.2

Report Date: February 17, 2004

Updated: July 12, 2004

Description	Part No.	Src	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
IBM eServer pSeries 690	7040-681	1	13,339	1	13,339	20,760
16X/48X(MAX) IDE DVD-ROM DR	2634	1	275	1	275	0
Ext Cable, Diskette/Op Panel, CEC to Media (3155,3255)	3155	1	250	1	250	0
128MB (4x32) L3 Cache, 567MHz	4199	1	25,000	4	100,000	0
SCSI-To-IDE Interface Bridge	4253	1	100	1	100	0
128GB Memory, 633 MHz, Inward/outward (4470,4471)	4470	1	284,426	8	2,275,408	0
8-Way Power4+Turbo 1.9 GHZ Processor	5241	1	380,000	4	1,520,000	420,000
Programmable Clock Card	5251	1	350	1	350	0
Std CEC Fan Pwr r cbl grp, service proc. 6161,6162	6161	1	600	1	600	0
Standard CEC DC/DC Converter Assembly	6170	1	4,700	2	9,400	0
Pow er Cable Grp, 1st-4th (6181,82,83,84) proc. modules	6181	1	720	1	720	0
Additional DC/DC Converter Assembly, (DCA)	6189	1	4,200	4	16,800	0
Capacitor book (for 2 MCMs)	6198	1	1,800	2	3,600	0
Service Processor + Dual Rio Loop (GX)	6418	1	7,000	1	7,000	0
Remote I/O Loop adapter	6419	1	8,000	1	8,000	0
Backplane, Central Electronics Complex	6565	1	50,000	1	50,000	0
Attachment Cable, Hardware Management Console	8121	1	75	1	75	0
Media Drw r, 1U, Op Pnl/Dskt/4 SCSI Media Bays	8692	1	1,700	1	1,700	0
I/O Draw er	7040-61D	1	3,980	4	15,920	16,320
Converter Cable, VHDCI to P, 0.3M	2118	1	60	4	240	0
SCSI cable - B&C to Media Draw er	2122	1	275	1	275	0
Fibre Chan. Converter Cable, LC (M) to SC (F), 2M	2456	1	100	4	400	0
RIO-G Cable, 500MHz, 1.2m	3146	1	500	4	2,000	0
RIO-G Cable, 500MHz, 1.75m	3156	1	375	8	3,000	0
36.4 GB 10,000 RPM Ultra3 SCSI Disk Drive	3158	1	1,540	5	7,700	0
GIGABIT ETHERNET-SX PCI-X ADAP	5700	1	1,700	1	1,700	0
PCI-X D.CHAN.U320 BLIND SW.ADP	5710	1	790	1	790	0
Cable Grp, (6121, 6122, 6123)	6121	1	400	3	1,200	0
I/O Drw r DC/DC Converter assembly, (DCA)	6172	1	4000	8	32,000	0
Pow er Cable, B&C to Media Draw er	6179	1	300	1	300	0
2 Gigabit Fibre Channel PCI-X Adapter	6239	1	3,100	28	86,800	0
Ultra3 SCSI 4-Pack	6564	1	500	16	8,000	0
I/O Draw er PCI-X Planar, 10 Slot, 2 Integrated	6571	1	8,000	8	64,000	0
Rack	7040-61R	1	5,500	1	5,500	3,960
Front Door, 24W, 2M, IBM Black/Copper	6070	1	3000	1	3,000	0
Rear Door, Slim line, Prim/sec.	6074	1	750	1	750	0
Bulk Pow er Regulator (BPR)	6186	1	4000	6	24,000	0
Bulk Pow er Controller (BPC), 4 Fans + 3 DCAs	6187	1	1900	2	3,800	0
Bulk Pow er Distribution (BPD), 10 DCAs	6188	1	3500	4	14,000	0
Line Cord, 60A, 14', IEC309 Plug, Chargeable	8678	1	1000	2	2,000	0
Regatta-H Bulk Pow er Module (2x BPA, EPO PNL)	8690	1	5000	1	5,000	0
Hardw are Management Console, mouse, keyboard, monitor, c	7315-C03	1	4486	1	4,486	696
				Subtotal	4,294,478	461,736
FASt Storage (Data Disks)						
RS/6000 System Rack (6081,6098)	7014-T42	1	4,270	14	59,780	12,432
FASt EXP 700	1740-1RU	1	6,000	147	882,000	111,720
Short Wave SFP	2210	1	499	588	293,412	0

2Gb FC, 36.4GB/15K Drive	5212	1	1,115	1,950	2,174,250	0	
2GB FC 73.4 GB/15K Disk Module	5213	1	2,225	40	89,000	0	
FAStT900 Storage Server	1742-90U	1	66,500	14	931,000	15,218	
Short Wave SFP GBIC	2210	1	499	82	40,918	0	
Fiber Cable 25m	5625	1	189	28	5,292	0	
Fiber Cable 1m	5601	1	79	268	21,172	0	
Fiber Cable 5m	5605	1	129	26	3,354	0	
FAStT Storage Manager V8.4 upgrade for FAStT900	1742-7109	1	2,999	1	2,999	0	
FAStT700 2 gbps Mini HUB	3507	1	899	56	50,344	0	
			Subtotal		4,553,521	139,370	
Server Software							
AIX 5.2 (media only)	5962-A5L	1	50	1	50	0	
AIX SWMA Subscription (3 Years)	5773-SUB	1	743	32	23,776	0	
AIX SWMA Support (0280 + 0382) (3 Years)	5773-SPT	1	2,093	32	0	66,976	
HMC Softw are SUB	5639-SB1	1	275	3	825		
HMC Softw are Support	5639-ST1	1	525	3		1,575	
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	515	1	515	0	
C for AIX user annual SW maint renew al	E1A1FLL	1	103	2	0	206	
DB2 Enterprise Server Edition Proc Lic/1 yr Maint.	D518GLL	1	19,755	32	632,160	0	
DB2 Enterprise Server Ed Proc Maint Renew (annual).	E00BILL	1	940	64		60,160	
			Subtotal		657,326	128,917	
Client Hardware and Software							
x225 with 2.8GHz/512KB Xeon DP, 512GB memory	8647-5AX	1	1,399	40	55,960	27,920	
2.8GHz/512KB Zeon DP Processor Upgrade	24P8054	1	799	40	31,960	0	
512MB PC2100 CL2.5 ECC DDR SDRAM RDIMM	33L5038	1	275	160	44,000	0	
IBM ServicePac for xSeries	29R5396	1	1,375	40	0	55,000	
36.4 GB 10K Ultra 320 SCSI Drive	32P0726	1	275	40	11,000	0	
NetXtreme 1000 T Dual-Port Ethernet Adaptor	31P6401	1	269	40	10,760	0	
IBM Thinkvision L200p 20.1" LCD color monitor	6736HB0	1	1,349	1	1,349	0	
			Subtotal		155,029	82,920	
Third Party Hardware/Software							
Microsoft Window s 2000 server		2	999	40	39,960	0	
Visual Studio .NET Professional		2	1,079	1	1,079	0	
Adaptec Quartet66 4-port adaptor	1932500	3	550	40	22,000	0	
Netgear Managed Stackable Sw tich w/Gigabit ports -48 ports FSM750SNA		4	759	1	759	0	
			Subtotal		63,798	0	
			Discounts		(4,679,313)	(292,009)	
			Total		5,044,838	520,934	

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

Pricing Sources:

	Three-Year Cost of Ownership	\$5,565,772
1 IBM HW: Bill Casey, eServer pSeries Offering Manager, wr Casey@us.ibm.com, 512-838-1422	tpm C	1,025,486
1 IBM DB2: Paul Rivot, Director Database Servers and Business Intelligence Software, e-mail privot@us.ibm.com, phone 1-914-766-1325	\$/tpm C	\$5.43
2. Microsoft.com		
3. Adaptec.com		
4. Insight.com		

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

Numerical Quantities Summary for the IBM eServer pSeries 690 Model 7040-681

MQTH, computed Maximum Qualified Throughput: 1,025,486.07 tpmC

<u>Response Times (in seconds)</u>	<u>90th %</u>	<u>Average</u>	<u>Maximum</u>
New Order	0.50	0.32	3.15
Payment	0.51	0.33	3.33
Order-Status	0.50	0.32	2.65
Delivery (interactive)	0.29	0.13	2.06
Delivery (deferred)	0.27	0.17	1.16
Stock-Level	0.49	0.31	2.79
Menu	0.32	0.15	7.55

<u>Transaction Mix, in percent of total transactions</u>	<u>Percent</u>
New Order	44.91%
Payment	43.02%
Order-Status	4.02%
Delivery	4.01%
Stock-Level	4.01%

<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	19.12/120.30
Payment	3.00/0.01	3.01/12.02	4.56/120.30
Order-Status	2.00/0.01	2.01/10.01	2.81/100.16
Delivery	2.00/0.01	2.01/5.03	2.95/50.26
Stock-Level	2.00/0.01	2.01/5.02	2.91/50.29

Test Duration

Ramp-up Time	58 minutes
Measurement interval	2 hours
Transactions during measurement interval (all types)	273,998,416
Ramp-down time	20 minutes

Checkpoints

Number of checkpoints	N/A
Checkpoint interval	N/A

Approximately two months after the publication this result was found to be in insignificant deviation with TPC-C Version 5.2 Specification. The driver (RTE) did not include the required 0.1-second delay to compensate for the time taken by the web browser to display each user screen. Evidence indicates that the increase in transaction cycle time, without changing any other aspect of the benchmark, would degrade the performance and price / performance metrics by approximately one percent. An amended auditor attestation letter has been attached to the FDR.

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.2 dated December, 2003, for measurements on the IBM eServer pSeries 690 Model 7040-681. The software used on the IBM eServer pSeries 690 Model 7040-681 includes AIX 5L Version 5.2 operating system, DB2 UDB 8.1 database manager. Microsoft COM+ is used as transaction manager.

IBM eServer pSeries 690 Model 7040-681

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM eServer pSeries 690 Model 7040-681	DB2 UDB 8.1	AIX 5L Version 5.2

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
5,565,772	1,025,486.07	\$5.43 per tpmC

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 5.2 in December 2003.

This is the full disclosure report for benchmark testing of the IBM eServer pSeries 690 Model 7040-681 and DB2 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.

1. General Items

1.1 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.

1.2 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**.

1.3 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.*

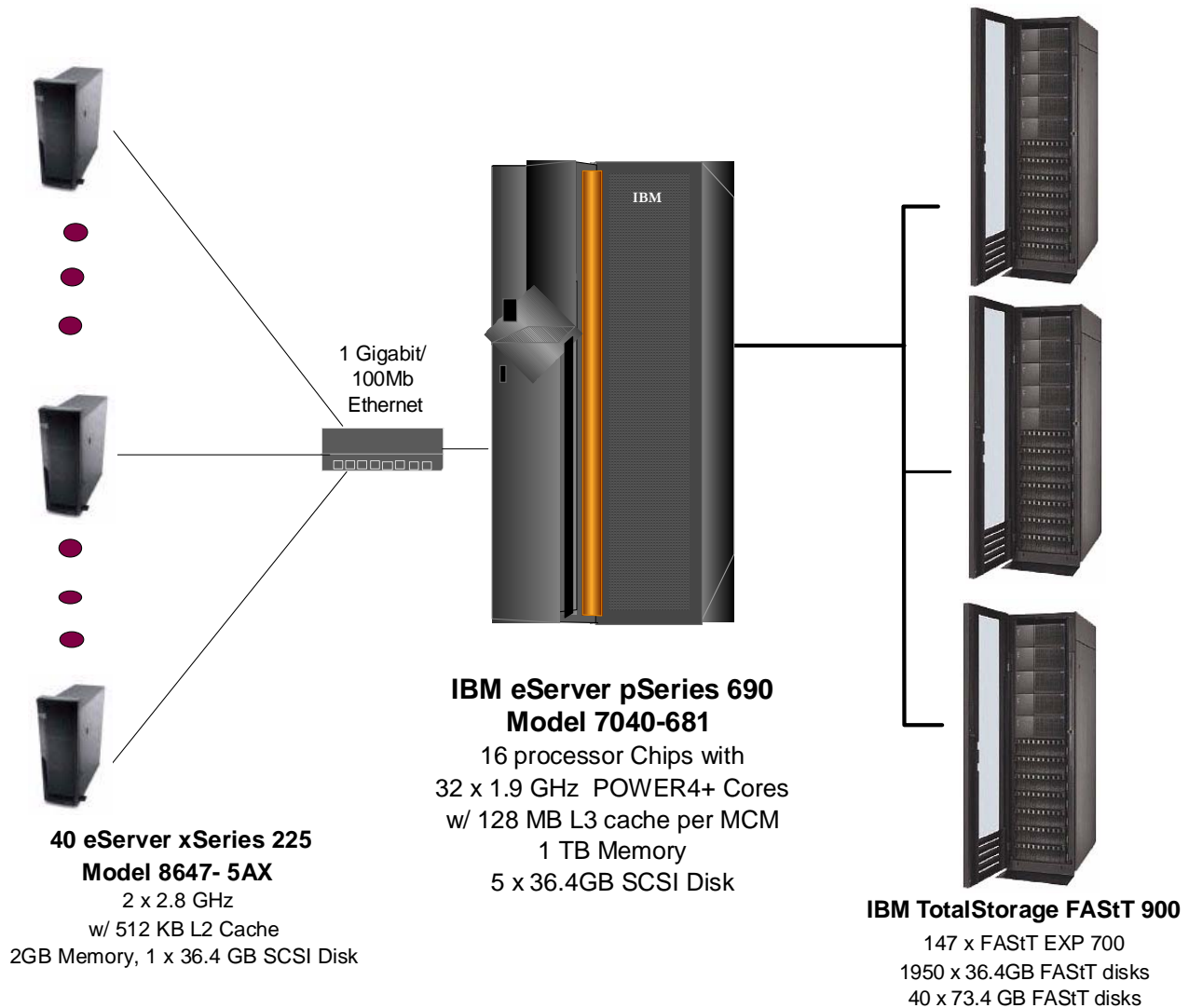
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

1.4 Configuration Diagrams

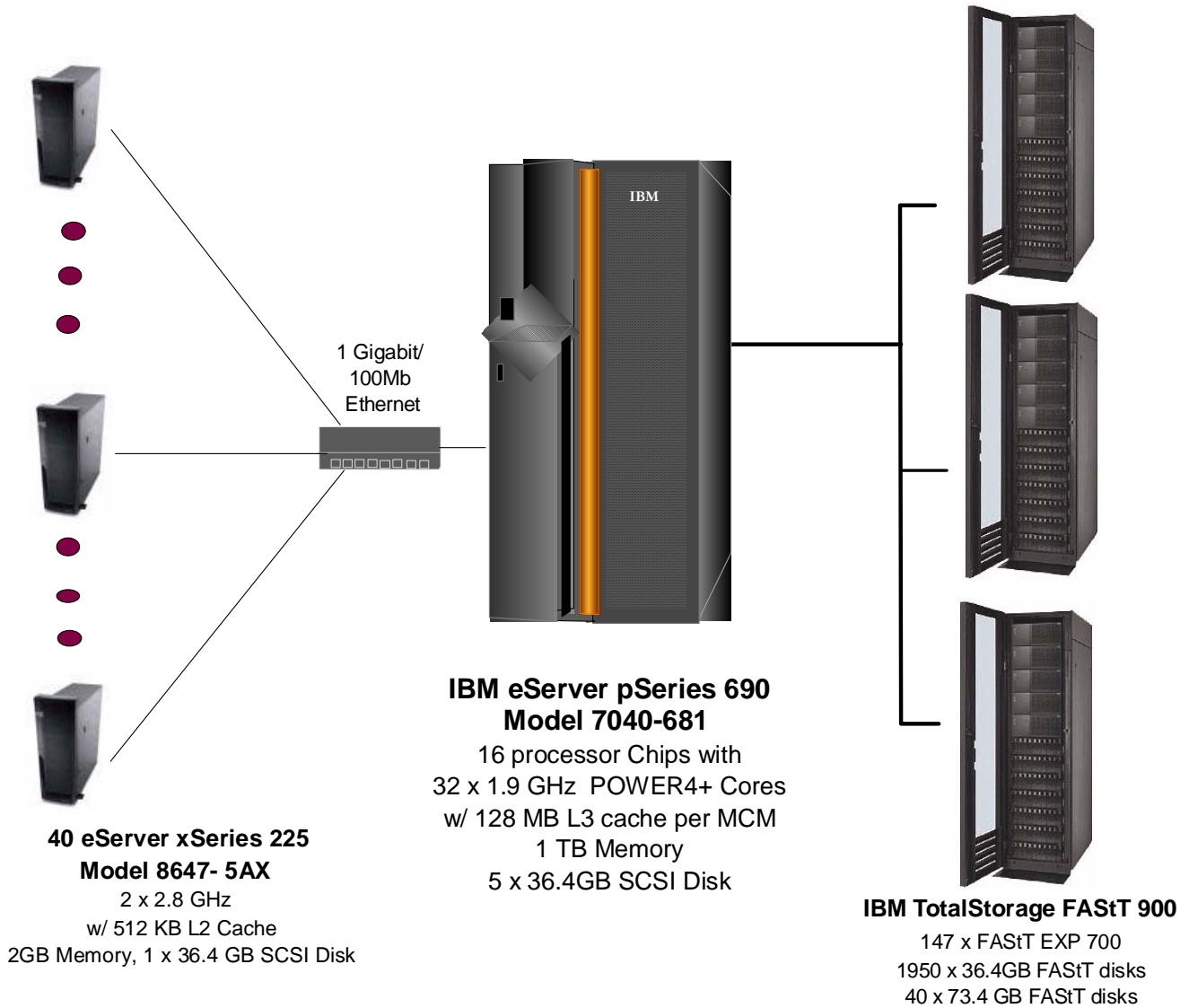
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *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 pSeries 690 Model 7040-681 Benchmark Configuration



IBM eServer pSeries 690 Model 7040-681 Priced Configuration



2. Clause 1: Logical Data Base Design Related Items

2.1 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.

2.2 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.

2.3 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.

2.4 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.

The tables Warehouse, District, Customer, Order Line, Order, New Order, History, and Stock were horizontally partitioned into multiple tables. Each table partition had 2150 warehouses, except for Stock, which had 1075 warehouses per partition. No tables were replicated. A view was created over all partitioned tables to provide full transparency of data manipulation.

3. Clause 2: Transaction and Terminal Profiles Related Items

3.1 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.

3.2 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.

3.3 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 xSeries Model 8647-5AX, are commercially available and support all of the requirements in Clause 2.2.2.4.

3.4 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.

3.5 Home and Remote Order-lines

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

Table 3-1 shows the percentage of home and remote transactions that occurred during the measurement period for the New-Order transactions.

3.6 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 3-1 shows the percentage of New-Order transactions that were rolled back due to an illegal item being entered.

3.7 Number of Items per Order

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

Table 3-1 show the average number of items ordered per New-Order transaction.

3.8 Home and Remote Payment Transactions

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

Table 3-1 shows the percentage of home and remote transactions that occurred during the measurement period for the Payment transactions.

3.9 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 3-1 shows the percentage of non-primary key accesses to the data base by the Payment and Order-Status transactions.

3.10 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 3-1 shows the percentage of Delivery transactions missed due to a shortage of supply of rows in the NEW-ORDER table.

3.11 Mix of Transaction Types

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

Table 3-1 shows the mix percentage for each of the transaction types executed by the SUT.

3.12 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.

Table 3-1 Numerical Quantities for Transaction and Terminal Profiles

	IBM eServer pSeries 690 Model 7040-681
New Order	
Percentage of Home order lines	99.00%
Percentage of Remote order lines	1.00%
Percentage of Rolled Back Transactions	1.00%
Average Number of Items per order	9.99
Payment	
Percentage of Home transactions	85.01%
Percentage of Remote transactions	14.99%
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	44.91%
Payment	43.02%
Order-Status	4.02%
Delivery	4.01%
Stock-Level	4.01%

4. 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.

4.1 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.

4.1.1 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.
-

4.1.2 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.

4.2 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:

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

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.

4.3 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.

4.4 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

4.4.1 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-10 (mirrored), 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 about 20% 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.

5. Clause 4: Scaling and Data Base Population Related Items

5.1 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 5-1 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 81700 warehouses, the number of active warehouses during the benchmark.

Table 5-1 Initial Cardinality of Tables (IBM eServer pSeries 690 Model 7040-681).

Table Name	Number of Rows
Warehouse	81,700
District	817,000
Customer	2,451,000,000
History	2,451,000,000
Order	2,451,000,000
New Order	735,300,000
Order Line	24,509,846,400
Stock	8,170,000,000
Item	100,000

5.2 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 are 2 storage adapters for the logs. Each adapter has 2 logical disks (hdisks) and each hdisk has 10 physical disks, for a total of 40 log disks. The hdisks are configured as RAID10 disk arrays. Each physical disk has a capacity of 73.4 GB. The log LV is striped across the 4 hdisks.

There are 26 storage adapters for the data disks. Each adapter has 3 hdisks and each hdisk has 25 physical disks, for a total of 1950 data disks. Each hdisk is configured as a RAID0 storage array. The physical disks are 36.4 GB drives.

Each data logical volume (LV) is distributed across 2 hdisks. The 2 hdisks are on separate adapters. There are a total of 39 LVs per adapter pair.

Each partition within a partitioned table is made of one DB2 container (Logical Volume) so that the corresponding view spans 26 adapters. The ITEM table, which is not partitioned, is made of 39 containers and also spans 26 adapters.

Also, for each partitioned table, each adapter pair has 3 logical volumes representing 3 partitions, except for Stock which has 6 logical volumes representing 6 stock partitions.

Disk Mapping:

Table 5-2: IBM eServer pSeries 690 Model 7040-681 Data Distribution Benchmark Configuration

hdisk	Adapter	Logical Volumes
hdisk11 hdisk16	fcs41	db2loglv
hdisk13 hdisk17	fcs42	
hdisk18	fcs44	D1F01V1ITEM, D1F01V1WARE, D1F01V1DIST, D1F01V1NORI, D1F01V1CUST, D1F01ORL, D1F01V1STK, D1F01V1CSTI, D1F01V1ORD, D1F01V1NOR, D1F01V1HIST, D1F01V1ORDI
hdisk19	fcs43	
hdisk20	fcs44	D1F01V2ITEM, D1F01V2WARE, D1F01V2DIST, D1F01V2NORI, D1F01V2CUST, D1F01V2ORL, D1F01V2STK, D1F01V2CSTI, D1F01V2ORD, D1F01V2NOR, D1F01V2HIST, D1F01V2ORDI
hdisk21	fcs43	
hdisk22	fcs44	D1F01V3ITEM, D1F01V3WARE, D1F01V3DIST, D1F01V3NORI, D1F01V3CUST, D1F01V3ORL, D1F01V3STK, D1F01V3CSTI, D1F01V3ORD, D1F01V3NOR, D1F01V3HIST, D1F01V3ORDI
hdisk23	fcs43	
hdisk113	fcs2	D1F02V1ITEM, D1F02V1WARE, D1F02V1DIST, D1F02V1NORI, D1F02V1CUST, D1F02V1ORL, D1F02V1STK, D1F02V1CSTI, D1F02V1ORD, D1F02V1NOR, D1F02V1HIST, D1F02V1ORDI
hdisk114	fcs1	
hdisk126	fcs2	D1F02V2ITEM, D1F02V2WARE, D1F02V2DIST, D1F02V2NORI, D1F02V2CUST, D1F02V2ORL, D1F02V2STK, D1F02V2CSTI, D1F02V2ORD, D1F02V2NOR, D1F02V2HIST, D1F02V2ORDI
hdisk127	fcs1	
hdisk128	fcs1	D1F02V3ITEM, D1F02V3WARE, D1F02V3DIST, D1F02V3NORI, D1F02V3CUST, D1F02V3ORL, D1F02V3STK, D1F02V3CSTI, D1F02V3ORD, D1F02V3NOR, D1F02V3HIST, D1F02V3ORDI
hdisk129	fcs2	
hdisk42	fcs54	D1F03V1ITEM, D1F03V1WARE, D1F03V1DIST, D1F03V1NORI, D1F03V1CUST, D1F03V1ORL, D1F03V1STK, D1F03V1CSTI, D1F03V1ORD, D1F03V1NOR, D1F03V1HIST, D1F03V1ORDI
hdisk43	fcs53	
hdisk44	fcs54	D1F03V2ITEM, D1F03V2WARE, D1F03V2DIST, D1F03V2NORI, D1F03V2CUST, D1F03V2ORL, D1F03V2STK, D1F03V2CSTI, D1F03V2ORD, D1F03V2NOR, D1F03V2HIST, D1F03V2ORDI
hdisk45	fcs53	
hdisk46	fcs54	D1F03V3ITEM, D1F03V3WARE, D1F03V3DIST, D1F03V3NORI, D1F03V3CUST, D1F03V3ORL, D1F03V3STK, D1F03V3CSTI, D1F03V3ORD, D1F03V3NOR, D1F03V3HIST, D1F03V3ORDI
hdisk47	fcs53	
hdisk54	fcs57	D1F04V1ITEM, D1F04V1WARE, D1F04V1DIST, D1F04V1NORI, D1F04V1CUST, D1F04V1ORL, D1F04V1STK, D1F04V1CSTI, D1F04V1ORD, D1F04V1NOR, D1F04V1HIST, D1F04V1ORDI
hdisk55	fcs56	
hdisk56	fcs57	D1F04V2ITEM, D1F04V2WARE, D1F04V2DIST, D1F04V2NORI, D1F04V2CUST, D1F04V2ORL, D1F04V2STK, D1F04V2CSTI, D1F04V2ORD, D1F04V2NOR, D1F04V2HIST, D1F03V2ORDI
hdisk57	fcs56	
hdisk58	fcs57	D1F04V3ITEM, D1F04V3WARE, D1F04V3DIST, D1F04V3NORI, D1F04V3CUST, D1F04V3ORL, D1F04V3STK, D1F04V3CSTI, D1F04V3ORD, D1F04V3NOR, D1F04V3HIST, D1F04V3ORDI
hdisk59	fcs56	
hdisk60	fcs3	D1F05V1ITEM, D1F05V1WARE, D1F05V1DIST, D1F05V1NORI, D1F05V1CUST, D1F05V1ORL, D1F05V1STK, D1F05V1CSTI, D1F05V1ORD, D1F05V1NOR, D1F05V1HIST, D1F05V1ORDI
hdisk61	fcs58	
hdisk62	fcs3	D1F05V2ITEM, D1F05V2WARE, D1F05V2DIST, D1F05V2NORI, D1F05V2CUST, D1F05V2ORL, D1F05V2STK, D1F05V2CSTI, D1F05V2ORD, D1F05V2NOR, D1F05V2HIST, D1F05V2ORDI
hdisk74	fcs58	
hdisk75	fcs3	D1F05V3ITEM, D1F05V3WARE, D1F05V3DIST, D1F05V3NORI, D1F05V3CUST, D1F05V3ORL, D1F05V3STK, D1F05V3CSTI, D1F05V3ORD, D1F05V3NOR, D1F05V3HIST, D1F05V3ORDI
hdisk76	fcs58	
hdisk77	fcs61	D1F06V1ITEM, D1F06V1WARE, D1F06V1DIST, D1F06V1NORI, D1F06V1CUST, D1F06V1ORL, D1F06V1STK, D1F06V1CSTI, D1F06V1ORD, D1F06V1NOR, D1F06V1HIST, D1F06V1ORDI
hdisk78	fcs60	
hdisk79	fcs61	D1F06V2ITEM, D1F06V2WARE, D1F06V2DIST, D1F06V2NORI, D1F06V2CUST, D1F06V2ORL, D1F06V2STK,

hdisk80	fcs60	D1F06V2CSTI, D1F06V2ORD, D1F06V1NOR, D1F06V2HIST, D1F06V2ORDI
hdisk81	fcs61	D1F06V3ITEM, D1F06V3WARE, D1F06V3DIST, D1F06V3NORI, D1F06V3CUST, D1F06V3ORL, D1F06V3STK, D1F06V3CSTI, D1F06V3ORD, D1F06V3NOR, D1F06V3HIST, D1F06V3ORDI
hdisk82	fcs60	
hdisk83	fcs63	D1F07V1ITEM, D1F07V1WARE, D1F07V1DIST, D1F07V1NORI, D1F07V1CUST, D1F07V1ORL, D1F07V1STK, D1F07V1CSTI, D1F07V1ORD, D1F07V1NOR, D1F07V1HIST, D1F07V1ORDI
hdisk84	fcs62	
hdisk85	fcs63	D1F07V2ITEM, D1F07V2WARE, D1F07V2DIST, D1F07V2NORI, D1F07V2CUST, D1F07V2ORL, D1F07V2STK, D1F07V2CSTI, D1F07V2ORD, D1F07V2NOR, D1F07V2HIST, D1F07V2ORDI
hdisk86	fcs62	
hdisk87	fcs63	D1F07V3ITEM, D1F07V3WARE, D1F07V3DIST, D1F07V3NORI, D1F07V3CUST, D1F07V3ORL, D1F07V3STK, D1F07V3CSTI, D1F07V3ORD, D1F07V3NOR, D1F07V3HIST, D1F07V3ORDI
hdisk88	fcs62	
hdisk48	fcs0	D1F08V1ITEM, D1F08V1WARE, D1F08V1DIST, D1F08V1NORI, D1F08V1CUST, D1F08V1ORL, D1F08V1STK, D1F08V1CSTI, D1F08V1ORD, D1F08V1NOR, D1F08V1HIST, D1F08V1ORDI
hdisk49	fcs4	
hdisk50	fcs0	D1F08V2ITEM, D1F08V2WARE, D1F08V2DIST, D1F08V2NORI, D1F08V2CUST, D1F08V2ORL, D1F08V2STK, D1F08V2CSTI, D1F08V2ORD, D1F08V2NOR, D1F08V2HIST, D1F08V2ORDI
hdisk51	fcs4	
hdisk52	fcs0	D1F08V3ITEM, D1F08V3WARE, D1F08V3DIST, D1F08V3NORI, D1F08V3CUST, D1F08V3ORL, D1F08V3STK, D1F08V3CSTI, D1F08V3ORD, D1F08V3NOR, D1F08V3HIST, D1F08V3ORDI
hdisk53	fcs4	
hdisk89	fcs66	D1F09V1ITEM, D1F09V1WARE, D1F09V1DIST, D1F09V1NORI, D1F09V1CUST, D1F09V1ORL, D1F09V1STK, D1F09V1CSTI, D1F09V1ORD, D1F09V1NOR, D1F09V1HIST, D1F09V1ORDI
hdisk90	fcs65	
hdisk91	fcs66	D1F09V2ITEM, D1F09V2WARE, D1F09V2DIST, D1F09V2NORI, D1F09V2CUST, D1F09V2ORL, D1F09V2STK, D1F09V2CSTI, D1F09V2ORD, D1F09V2NOR, D1F09V2HIST, D1F09V2ORDI
hdisk92	fcs65	
hdisk93	fcs66	D1F09V3ITEM, D1F09V3WARE, D1F09V3DIST, D1F09V3NORI, D1F09V3CUST, D1F09V3ORL, D1F09V3STK, D1F09V3CSTI, D1F09V3ORD, D1F09V3NOR, D1F09V3HIST, D1F09V3ORDI
hdisk94	fcs65	
hdisk95	fcs68	D1F10V1ITEM, D1F10V1WARE, D1F10V1DIST, D1F10V1NORI, D1F10V1CUST, D1F10V1ORL, D1F10V1STK, D1F10V1CSTI, D1F10V1ORD, D1F10V1NOR, D1F10V1HIST, D1F10V1ORDI
hdisk96	fcs67	
hdisk97	fcs68	D1F10V2ITEM, D1F10V2WARE, D1F10V2DIST, D1F10V2NORI, D1F10V2CUST, D1F10V2ORL, D1F10V2STK, D1F10V2CSTI, D1F10V2ORD, D1F10V2NOR, D1F10V2HIST, D1F10V2ORDI
hdisk98	fcs67	
hdisk99	fcs68	D1F10V3ITEM, D1F10V3WARE, D1F10V3DIST, D1F10V3NORI, D1F10V3CUST, D1F10V3ORL, D1F10V3STK, D1F10V3CSTI, D1F10V3ORD, D1F10V3NOR, D1F10V3HIST, D1F10V3ORDI
hdisk100	fcs67	
hdisk101	fcs71	D1F11V1ITEM, D1F11V1WARE, D1F11V1DIST, D1F11V1NORI, D1F11V1CUST, D1F11V1ORL, D1F11V1STK, D1F11V1CSTI, D1F11V1ORD, D1F11V1NOR, D1F11V1HIST, D1F11V1ORDI
hdisk102	fcs70	
hdisk103	fcs71	D1F11V2ITEM, D1F11V2WARE, D1F11V2DIST, D1F11V2NORI, D1F11V2CUST, D1F11V2ORL, D1F11V2STK, D1F11V2CSTI, D1F11V2ORD, D1F11V2NOR, D1F11V2HIST, D1F11V2ORDI
hdisk104	fcs70	
hdisk105	fcs71	D1F11V3ITEM, D1F11V3WARE, D1F11V3DIST, D1F11V3NORI, D1F11V3CUST, D1F11V3ORL, D1F11V3STK, D1F11V3CSTI, D1F11V3ORD, D1F11V3NOR, D1F11V3HIST, D1F11V3ORDI
hdisk106	fcs70	
hdisk107	fcs72	D1F12V1ITEM, D1F12V1WARE, D1F12V1DIST, D1F12V1NORI, D1F12V1CUST, D1F12V1ORL, D1F12V1STK, D1F12V1CSTI, D1F12V1ORD, D1F12V1NOR, D1F12V1HIST, D1F12V1ORDI
hdisk108	fcs73	
hdisk109	fcs72	D1F12V2ITEM, D1F12V2WARE, D1F12V2DIST, D1F12V2NORI, D1F12V2CUST, D1F12V2ORL, D1F12V2STK, D1F12V2CSTI, D1F12V2ORD, D1F12V2NOR, D1F12V2HIST, D1F12V2ORDI
hdisk110	fcs73	
hdisk111	fcs72	D1F12V3ITEM, D1F12V3WARE, D1F12V3DIST, D1F12V3NORI, D1F12V3CUST, D1F12V3ORL, D1F12V3STK, D1F12V3CSTI, D1F12V3ORD, D1F12V3NOR, D1F12V3HIST, D1F12V3ORDI
hdisk112	fcs73	

hdisk36	fcs52	D1F13V1ITEM, D1F13V1WARE, D1F13V1DIST, D1F13V1NORI, D1F13V1CUST, D1F13V1ORL, D1F10V1STK, D1F13V1CSTI, D1F13V1ORD, D1F13V1NOR, D1F13V1HIST, D1F13V1ORDI
hdisk37	fcs51	
hdisk38	fcs52	D1F13V2ITEM, D1F13V2WARE, D1F13V2DIST, D1F13V2NORI, D1F13V2CUST, D1F13V2ORL, D1F13V2STK, D1F13V2CSTI, D1F13V2ORD, D1F13V1NOR, D1F13V2HIST, D1F13V2ORDI
hdisk39	fcs51	
hdisk40	fcs52	D1F13V3ITEM, D1F13V3WARE, D1F13V3DIST, D1F13V3NORI, D1F13V3CUST, D1F13V3ORL, D1F13V3STK, D1F13V3CSTI, D1F13V3ORD, D1F13V3NOR, D1F13V3HIST, D1F13V3ORDI
hdisk41	fcs51	

5.3 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 8.1. 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.

5.4 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 5-2.

5.5 60-day space calculations

IBM eServer pSeries 690 Model 7040-681

60 Days Space Computation

All Data sizes are in MB unless
otherwise stated

Warehouses	81,700	tpmC	1,025,486.07	tpmC/W	12.55
Table	Rows	Data	Index	5% Space	Total Space
History	2,451,000,000	151,544	0	0	151,544
Orders	2,451,000,000	110,504	68,780	0	179,284
Order-line	24,509,846,404	2,957,084	0	0	2,957,084
New-order	735,300,000	25,194	0	0	25,194
Warehouse	81,700	19	0	1	20
District	817,000	133	0	7	140
Item	100,000	10	0	1	11
Customer	2,451,000,000	1,915,010	118,180	101,660	2,134,850
Stock	8,170,000,000	2,659,848	0	132,992	2,792,840
Free Space	867,105				
Dynamic Space	3,244,326				
Static Space	4,927,860				
Daily Growth	651,557				
Daily Spread	0				
60 Days MB	44,021,259				
60 Days GB	42,990				

Log File Storage Requirement:

Log Pages Written (KB)	74,053,964.00	log file storage used in a steady state 30 minute interval
Total N-O Txn	31,380,570.00	New Order transactions completed during the same 30 minute interval during steady state
Log per N-O txn	2.36	KB of log storage used per New Order transaction
8 Hour Log (GB)	1,129.00	8-hour log storage required

Disks Type	Disk Formatted Capacity (GB)	SUT # of Disks	SUT Capacity (GB)	Priced # of Disks	Priced Capacity (GB)
DB FastT 36.4GB	36.40	1,950	70,980	1,950	70,980
LOG FastT RAID10 (40 x 73.4GB)	73.40	40	2,936	40	2,936
OS SCSI 36GB	36.40	5		5	182
Priced Space					74,098

6. Clause 5: Performance Metrics and Response Time Related Items

6.1 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 6-1 lists the response times and the ninetieth percentiles for each of the transaction types for the measured system.

6.2 Keying and Think Times

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

Table 6-1 lists the TPC-C keying and think times for the measured system.

Table 6-1. IBM eServer pSeries 690 Model 7040-681 Response, Think and Keying Times

Response Times	New Order	Payment	Order Status	Delivery (int./def.)	Stock Level	Menus
90 %	0.50	0.51	0.50	0.29/0.27	0.49	0.32
Average	0.32	0.33	0.32	0.13/0.17	0.31	0.15
Maximum	3.15	3.33	2.65	2.06/1.16	2.79	7.55
		Think Times				
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.02	12.02	10.01	5.03	5.02	N/A
Maximum	120.30	120.30	100.16	50.26	50.29	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	19.12	4.56	2.81	2.95	2.91	N/A

6.3 Response Time Frequency Distribution

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

Figure 6-3-1. IBM eServer pSeries 690 Model 7040-681 New-Order Response Time Distribution

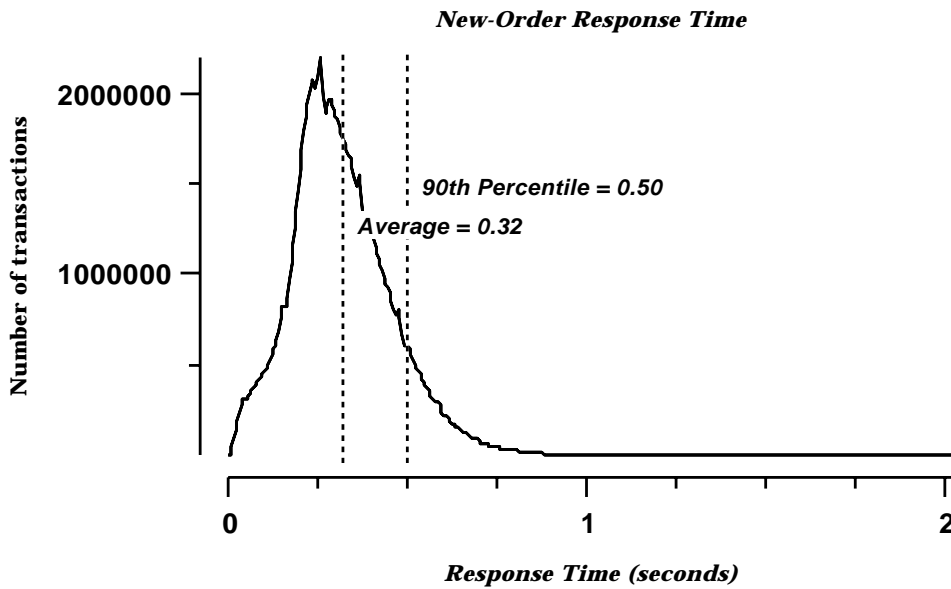


Figure 6-3-2. IBM eServer pSeries 690 Model 7040-681 Payment Response Time Distribution

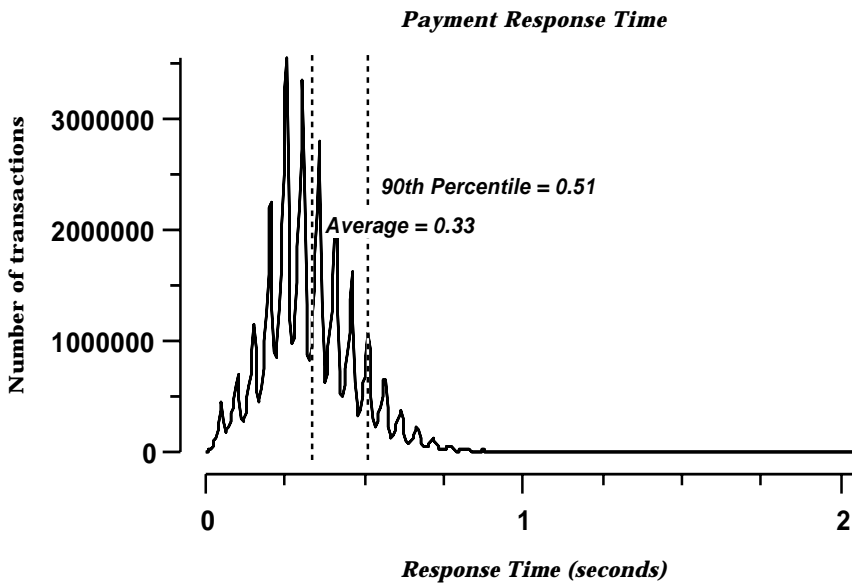


Figure 6-3-3. IBM eServer pSeries 690 Model 7040-681 Order-Status Response Time Distribution

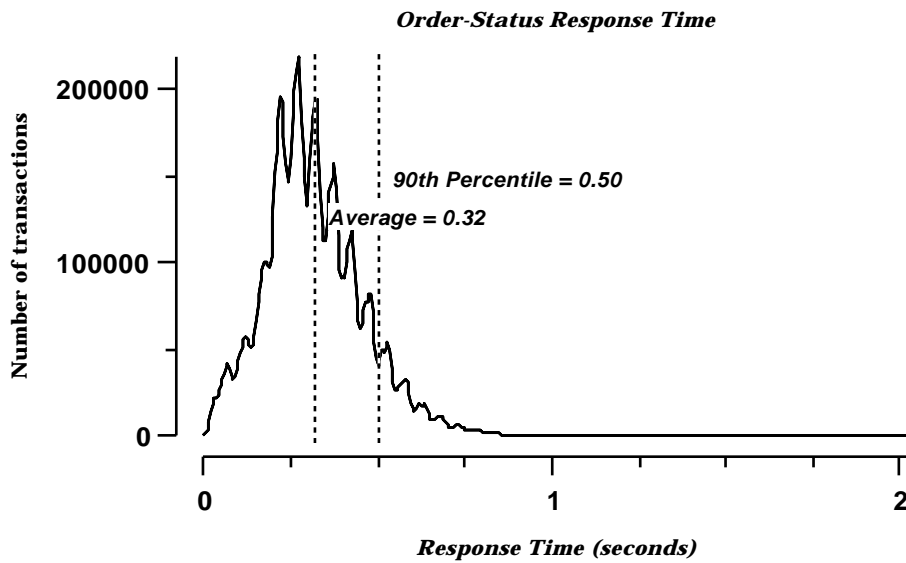


Figure 6-3-4. IBM eServer pSeries 690 Model 7040-681 Delivery (Interactive) Response Time Distribution

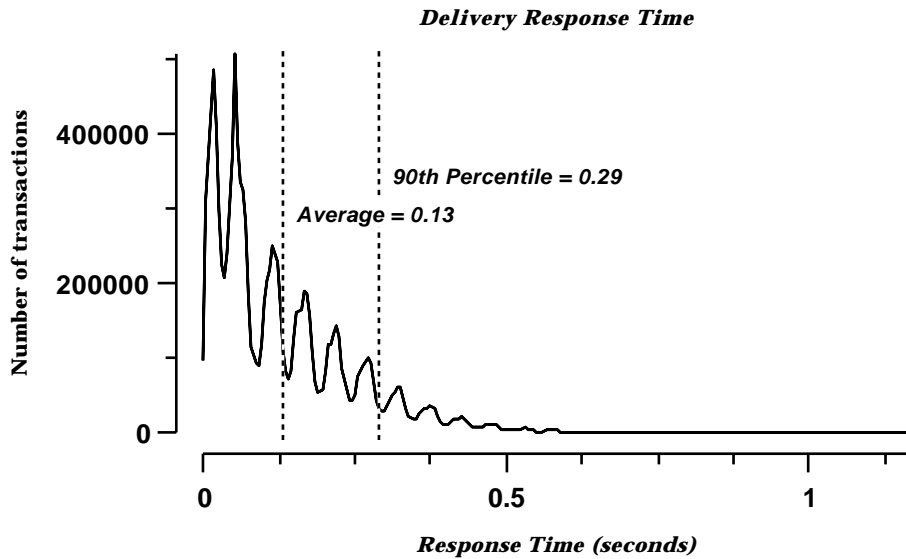


Figure 6-3-5. IBM eServer pSeries 690 Model 7040-681 Delivery (Deferred) Response Time Distribution

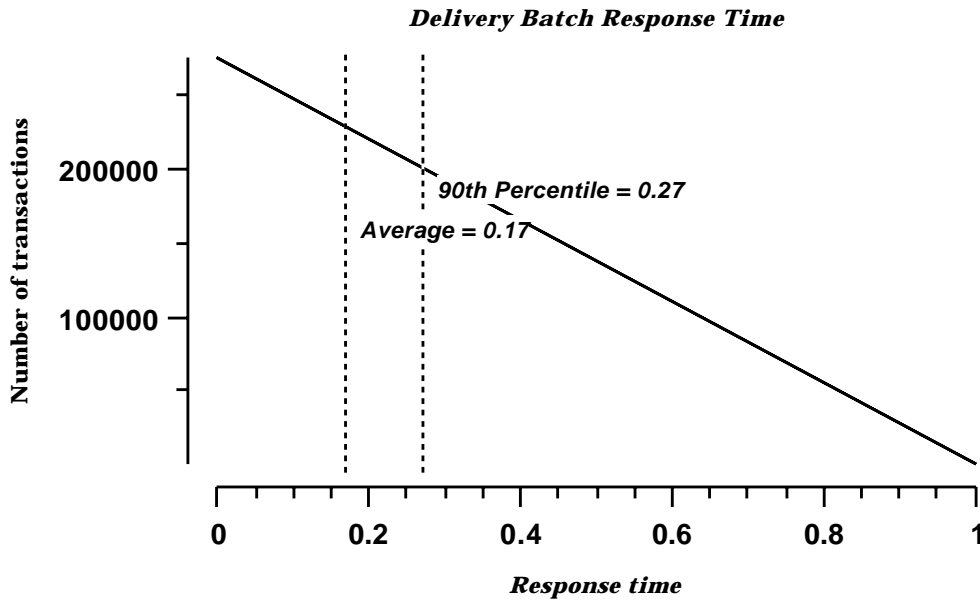
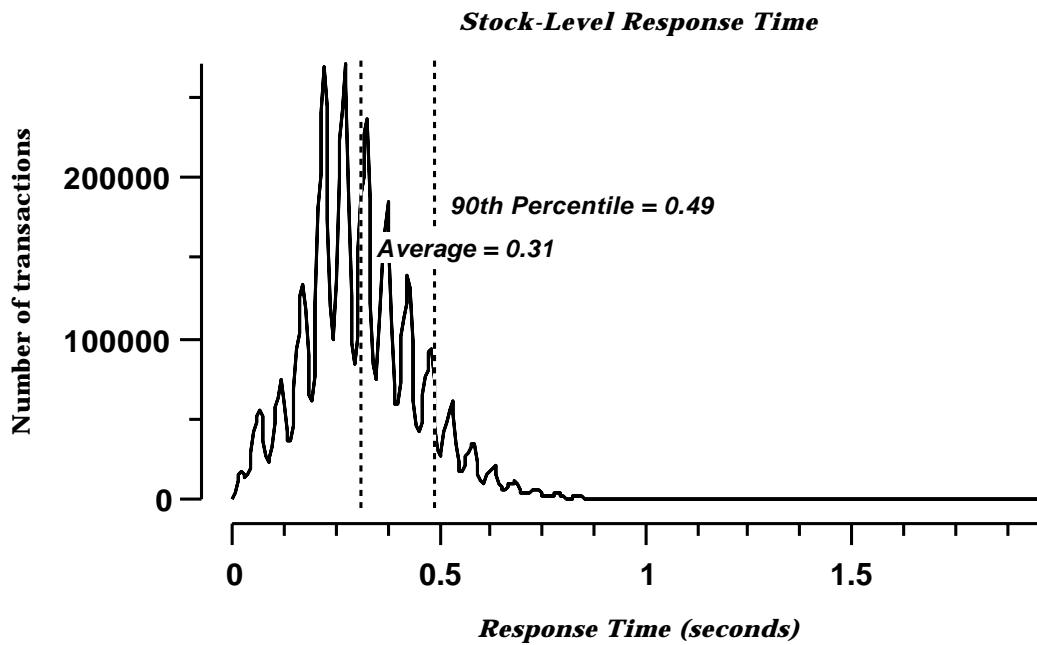


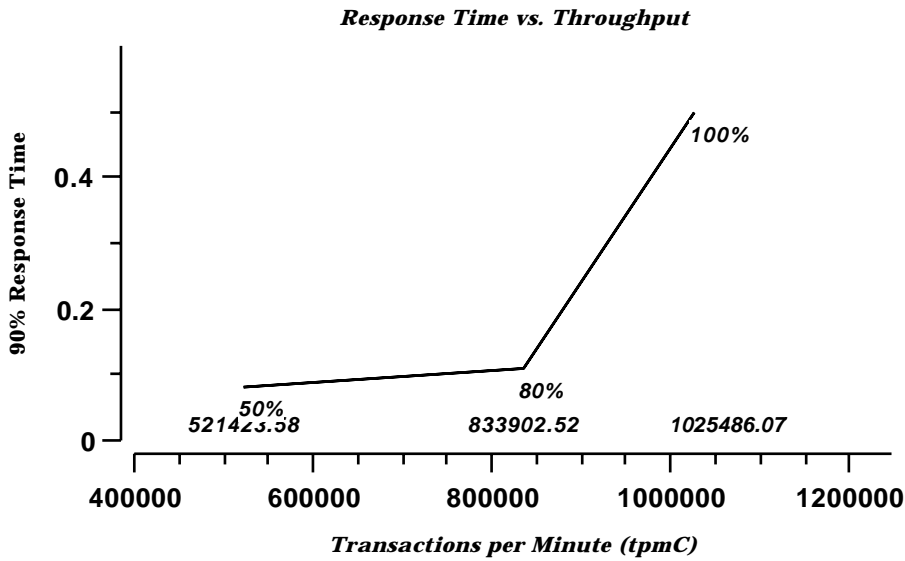
Figure 6-3-6. IBM eServer pSeries 690 Model 7040-681 Stock Level Response Time Distribution



6.4 Performance Curve for Response Time versus Throughput

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

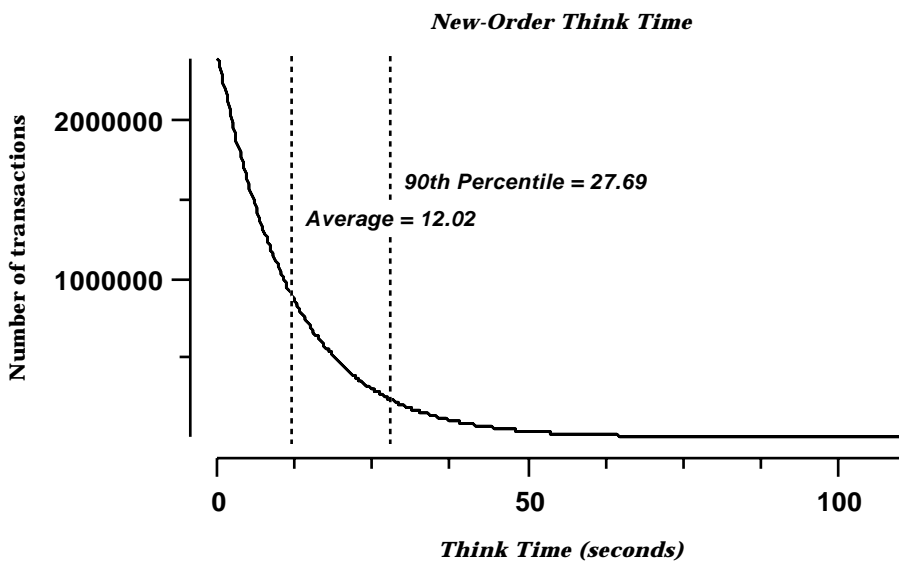
Figure 6-4-1. IBM eServer pSeries 690 Model 7040-681 New-Order Response Time vs. Throughput



6.5 Think Time Frequency Distribution

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

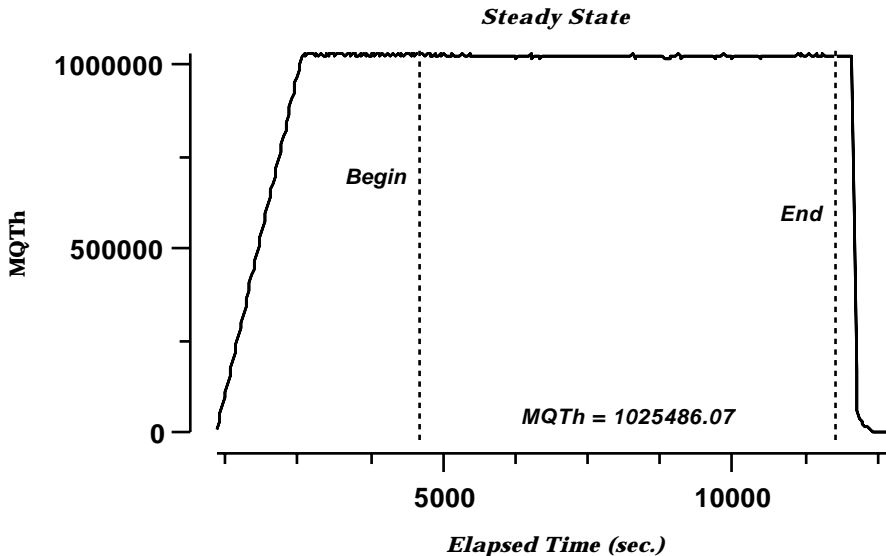
Figure 6-5-1. IBM eServer pSeries 690 Model 7040-681 New-Order Think Time Distribution



6.6 Throughput versus Elapsed Time

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

Figure 6-6-1. New-Order Throughput vs. Elapsed Time



6.7 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 6.6.1 New-Order throughput versus Elapsed Time graph shows that the system maintained a steady state during the measurement interval

6.8 Work Performed During Steady State

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

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

6.8.1 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.

6.8.2 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.

6.8.3 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))

6.9 Measurement Interval

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

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

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

7.1 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.

7.2 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.

In the benchmark configuration the Remote Terminal Emulator (RTE) communicates with the client system over Ethernet. The 50 eServer pSeries 630 Model 6E4 emulates a network of 817,000 eServer xSeries Model 8647-5AX workstations. The communications mechanism used in the benchmark and priced configurations are the same. In the benchmark configuration a two separate Ethernet LANs were used to connect one driver system to an eServer eServer xSeries Model 8647-5AX client system.

7.3 Network Bandwidth

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

The Ethernet used in the LAN complies with the IEEE 802.3 standard and has a bandwidth of 100 Mbits per second, Full Duplex. The clients are connected via 100 Mbit LANs to a 100 Mbit switch which then routes to a 1Gigabit ethernet which is connected to the SUT. LAN networks are capable of 100 Mbits per second are used in the priced configuration. The Gigabit Ethernet line connecting the clients and the SUT has a bandwidth of 1000 Mbits per second.

7.4 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.

8. Clause 7: Pricing Related Items

8.1 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.

8.2 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 price sheet for this disclosure is contained in the executive summary pages. The bases for the discounts used are:

3-year Term Maintenance Contract Discount

This discount is available for customers who sign a 3-year maintenance agreement on the hardware. A discount of 3% is available for customers when they sign a 3-year maintenance agreement.

Scope Incentive

A 2% discount is applied for the ServiceElect contract that combines hardware maintenance with one or more services, which in this pricing report the selected service is SupportLine.

3-year Maintenance Prepay

This is a discount for prepayment of maintenance costs. A discount of 10.36% is available for this configuration based on payment for three years maintenance at the time of purchase. This discount is applied to the balance after the 3-year term maintenance contract discount and Scope discounts applied.

Hardware Volume Revenue Discount

The IBM-supplied hardware and software is discounted by 48% from the list price, based on the dollar value of this configuration only.

For assistance with any of these prices or their applicability to any customer's requirements, please contact the following individuals:

Hardware:

William R. Casey,
IBM eServer pSeries Offering Manager
wrcasey@us.ibm.com
(512) 838-1422

DB2:

Paul Rivot,
Director, Database Servers and Business Intelligence Software
privot@us.ibm.com
(914) 766-1325

8.3 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 pSeries 690 Model 7040-681	August 16, 2004
AIX 5L V5.2	August 16, 2004
DB2 UDB 8.1.7	August 16, 2004

8.4 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 pSeries 690 Model 7040-681	1,025,486.07	\$5,565,772	\$5.43	August 16, 2004

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

. 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
----------	---	--

February 16, 2004

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

Platform:	IBM eServer pSeries 690 Model 7040-681 c/s
Operating system:	AIX 5L V5.2
Database Manager:	DB2 UDB 8.1
Transaction Manager:	Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	NewOrder Response Time - 90%	tpmC
Server: IBM eServer pSeries 690 Model 7040-681				
32 x POWER4+ (1.9 GHz)	1024 GB (128 MB L3 cache/MCM)	5 x 36.4GB SCSI 40 x 73.4GB FAStT 1950 x 36.4GB FAStT	0.50 Sec.	1,025,486.07
Forty (40) Clients: IBM eServer xSeries 225 Model 8647-5AX (Specification for each)				
2 x Intel Xeon (2.8 GHz)	2 GB (512 KB L2 cache/cpu)	1 x 36.4GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 5.2 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 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 hours.
- Continuous checkpoints were 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:

After the results were produced, it was discovered that the driver (RTE) did not include the required 0.1-second delay to compensate for the time taken by the web browser to display each user screen. It was established that the presence of this delay would not have a significant impact on the measured results (less than 1% decrease in throughput.)

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: TPC-C Application Source

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.
##
## I 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=c1.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
LDFLAGS_EXEC=
LDFLAGS_SHLIB=/DLL
LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpc.def
LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib /LIBPATH:"C:\Program
Files\Microsoft Visual Studio\VC98\Lib" db2api.lib winmm.lib
LDFLAGS_OUT=/OUT:

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

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

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

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 I 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 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=%HOME%\sqllib
set TPCC_RUNDATA=%HOME%\tpccdata
```

```
@REM TPCC Debug Configuration
@REM This is the path where all error and debug logs are placed.
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.
**
** I 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>
typedef __int16 int16_t;
typedef __int32 int32_t;
typedef __int64 int64_t;

#include "lval.h"

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

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

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

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

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*
*****
***** */
/* NURand Constants */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.
*/
/*
*****
***** */
#define C_C_LAST_RUN 88
#define C_C_LAST_LOAD 173
#define C_C_ID 319
#define C_OL_I_ID 3849

```

```

#define A_C_LAST 255
#define A_C_ID 1023
#define A_OL_I_ID 8191

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

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

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
    int16_t len;
    int16_t pad[SPGENERAL_PAD];
    struct in_items_struct {
        int32_t s_OL_I_ID;
        int32_t s_OL_SUPPLY_W_ID;
        int16_t s_OL_QUANTITY;
        int16_t pad1[3];
    } in_item[15];
    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_o_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 81700
#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.
**
** I 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 [ Addr + 4 - 0 - 1 = Addr+3];
      *a ^= *b; // sets *a to 0x6A
      *b ^= *a; // sets *b to 0x12
      *a ^= *b; // sets *a to 0x78
```

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

```
*****
*****/
```

```
void SwapEndian(void *Addr, int nb)
```

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

```
*a ^= *b;
*b ^= *a;
*a ^= *b;
```

```
}
}
#endif //SWAP_ENDIAN
```

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

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

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

```
#endif // __TPCCAPP_H
```

include/tpccdbg.h

```
/*
*****
** Licensed Materials – Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** I 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.
##
## I 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 CK040116 \
           NOLINEMACRO

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

CFLAGS = $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDES) \
         -D$SQLA_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.
**
** I 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.
*****
*****/

/*
 * tpccctx.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; }

strcpy(dbname,in_dbname,8);
if (strcmp(in_username,"") == 0)
{
    EXEC SQL CONNECT TO :dbname IN SHARE MODE;
} else {
    strcpy(username,in_username,128);
    strcpy(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", __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", __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", __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", __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", __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", __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", __LINE__, &sqlca);
        return SQLCODE;
    }

    return 0;
}

```

Srv.Common/tpccdbg.c

```

/*****
*****
** Licensed Materials – Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** I 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) {
            strncpy(debugPath, p, DEBUG_PATH_SIZE);
        } else {
            strcpy(debugPath, "C:\\temp");
        }
        strcat(debugPath, "\\");
    }
    debugInit = 1;
}

/*-----*/
/*      sqlerror      */
/*-----*/
void sqlerror(int tranType, char *msg, int ptat, 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 at point %2d (%s)\n", tranName, ptat, msg);
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 = ' ';
}
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=====
===== \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\n");
}

```



```

fprintf(debug_fp, "\t\n\n");
fclose(debug_fp);
}

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

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

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

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

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

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

    fprintf(debug_fp, "\n=====
=====");

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

    fprintf(debug_fp, "out_ordstat_struct {\n"};
    fprintf(debug_fp, "ts_C_ID = %d (%X)\n",
            ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);
    fprintf(debug_fp, "ts_C_FIRST = %s\n",
            ordstat_ptr->s_C_FIRST);
    fprintf(debug_fp, "ts_C_MIDDLE = %s\n",
            ordstat_ptr->s_C_MIDDLE);
    fprintf(debug_fp, "ts_C_LAST = %s\n",
            ordstat_ptr->s_C_LAST);
    fprintf(debug_fp, "ts_C_BALANCE = %lld\n",
            ordstat_ptr->s_C_BALANCE);
    fprintf(debug_fp, "ts_O_ID = %d (%X)\n",
            ordstat_ptr->s_O_ID, ordstat_ptr->s_O_ID);

```

```

fprintf(debug_fp, "ts_O_ENTRY_D = %lld (%IX)\n",
        ordstat_ptr->s_O_ENTRY_D_time, ordstat_ptr-
->s_O_ENTRY_D_time);
fprintf(debug_fp, "ts_O_CARRIER_ID = %d (%X)\n",
        ordstat_ptr->s_O_CARRIER_ID, ordstat_ptr->s_O_CARRIER_ID);
fprintf(debug_fp, "ts_ol_cnt = %d (%X)\n",
        ordstat_ptr->s_ol_cnt, ordstat_ptr->s_ol_cnt);
fprintf(debug_fp, "ts_transtatus = %d (%X)\n",
        ordstat_ptr->s_transtatus, ordstat_ptr->s_transtatus);
fprintf(debug_fp, "tdeadlocks = %d (%X)\n",
        ordstat_ptr->deadlocks, ordstat_ptr->deadlocks);

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

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

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

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

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

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

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

```

```

fprintf(debug_fp, "\n=====
===== \n");

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

fprintf(debug_fp, "out_payment_struct {\n"};
fprintf(debug_fp, "\ts_H_DATE = %lld (%lX)\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");
fclose(debug_fp);
}

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

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

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

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

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

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

    fprintf(debug_fp, "\n=====
===== \n");

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

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

```

```

fclose(debug_fp);
}

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

```

Srv.Common/tpccmisc.c

```

/*****
*****
** Licensed Materials – Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** I 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)
{
return(time(NULL));
}

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
struct timeval t;
gettimeofday(&t,NULL);
return (t.tv_sec + (double)t.tv_usec/(1000*1000));
}

```

Src.Cli/Makefile

```

====
## Licensed Materials – Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## I 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 CK040116 \
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)

#
#####
#####
# 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.
**
** I 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_newword->s_O_OL_CNT = itemIndex ;

// Sort the original array
qsort( in_newword->in_item, in_newword->s_O_OL_CNT
      , sizeof ( in_newword->in_item[ 0 ] )
      , itemComparison
      ) ;

// Now purge the duplicates.

actualItemIndex = -1 ;

for ( itemIndex = 0
      ; itemIndex < in_newword->s_O_OL_CNT
      ; itemIndex++ )
{
    // If duplicate, just increment the item entry order count

    if ( actualItemIndex >= 0
        && in_newword->in_item[ actualItemIndex ].s_OL_I_ID ==
in_newword->in_item[ itemIndex ].s_OL_I_ID
        && in_newword->in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID
== in_newword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID
        )
    {
        in_newword->in_item[ actualItemIndex ].s_OL_QUANTITY +=
in_newword->in_item[ itemIndex ].s_OL_QUANTITY ;
    }
    else
    {
        actualItemIndex ++ ;
        in_newword->in_item[ actualItemIndex ].s_OL_I_ID = in_newword-
>in_item[ itemIndex ].s_OL_I_ID ;
        in_newword->in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID =
in_newword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID ;
        in_newword->in_item[ actualItemIndex ].s_OL_QUANTITY =
in_newword->in_item[ itemIndex ].s_OL_QUANTITY ;
    }
}

in_newword->s_O_OL_CNT = actualItemIndex + 1 ;

pHostvarInput = (struct vc_new_in *) in_newword ;
pHostvarInput->len = sizeof(struct in_newword_struct) -
SPGENERAL_ADJUST ;

pHostvarOutput = (struct vc_new_out *) newword ;
pHostvarOutput->len = sizeof(struct out_newword_struct) -
SPGENERAL_ADJUST ;

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

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

```

```

    SWAP_BYTE(in_newword->duplicate_items);
#endif //SWAP_ENDIAN

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

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

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

if ( sqlca.sqlcode == 0 )
{
    double wtax = newword->s_W_TAX / 10000.0 ;
    double dtax = newword->s_D_TAX / 10000.0 ;
    double cdisc = newword->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 defered from the SP to the client has been.

    newword->s_total_amount = 0 ;

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

            newword->s_total_amount += newword-
>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.
    newword->s_total_amount *= factor;

```

```

}
else
{
    sqlerror( NEWORD_SQL, "NEW", __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

    // The 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
    // The 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_lim
                , :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
            )
        ) 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

```

```

                , :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_lim
                , :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

```

```

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

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

        sqlerror( PAYMENT_SQL, "PAY", __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", __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

    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_ol_cnt);
    SWAP_BYTE(ordstat->s_transtatus);
    SWAP_BYTE(ordstat->deadlocks);
    for (itemIndex=0; itemIndex<ordstat->s_ol_cnt; itemIndex++)
    {
        SWAP_BYTE(ordstat->item[ itemIndex ].s_OL_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", __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", __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

    ) 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" , __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", __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_ordr(struct ordr_wrapper *ordr,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 "..\tpcc\api\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(pymt->out_paym.s_D_CITY,"Valley");
strcpy(pymt->out_paym.s_D_STATE,"TX");
strcpy(pymt->out_paym.s_D_ZIP,"78559");
strcpy(pymt->out_paym.s_C_FIRST,"Jim");
strcpy(pymt->out_paym.s_C_MIDDLE,"F");
strcpy(pymt->out_paym.s_C_LAST,"Truck");
strcpy(pymt->out_paym.s_C_STREET_1,"100 N Solis");

```



```

ords->out_ords.item[4].s_OL_QUANTITY = 5;
ords->out_ords.item[4].s_OL_DELIVERY_D_time = 1234567890;

dataSet = 0;
}

return OK;
}

extern "C" NULLDB_API int do_dlv(struct dlv_wrapper *dlvy,void *ctx)
{
dlvy->out_dlv.s_transtatus = 0;

if (dataSet == 0)
{
dataSet = 1;

for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
dlvy->out_dlv.s_O_ID[districtIndex]= 2055;
}
else
{
for(int districtIndex=0;districtIndex <
DISTRICTS_PER_WAREHOUSE;districtIndex++)
dlvy->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
#ifdef 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

#ifdef _WIN32_WINNT
#define _WIN32_WINNT 0x0403
#endif

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

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

// TODO: reference additional headers your program requires here

tpccIsapi/tpcc.h

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

// standard includes

#ifdef _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
////////////////////////////////////

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

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

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

#define DELIVERY_THREADS_NUM 100

#define DISTRICTS_PER_WAREHOUSE 10
////////////////////////////////////
// Transaction Codes
////////////////////////////////////

#define TXN_LOGIN 0
#define TXN_NEW_ORDER 1
#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:
%d\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_DATETIME_LEN 19
#define DEFAULT_DATE_LEN 11
#define DEFAULT_TIME_LEN 8

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

////////////////////////////////////
// String Field Lengths
////////////////////////////////////

#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    -1

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

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

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

#define ERR_INVALID_D_ID         "ERROR: Invalid District ID. Try Again."
#define ERR_INVALID_W_ID         "ERROR: Invalid Warehouse ID. Try Again."
#define ERR_INVALID_C_ID         "ERROR: Invalid Customer ID. Try Again."
#define ERR_INVALID_SUPPLY_W_ID  "ERROR: Invalid Item Supply Warehouse. Try Again."
#define ERR_INVALID_ITEM_NUM     "ERROR: Invalid Item Number. Try Again."
#define ERR_INVALID_ITEM_OTY     "ERROR: Invalid Item Qty. Try Again."
#define ERR_MISSING_C_ID_OR_CLAST "ERROR: Must Enter Customer Id or Customer Last Name. Try Again."
#define ERR_INVALID_PAYMENT_AMOUNT "ERROR: Invalid Payment Amount. Try Again."
#define ERR_INVALID_CARRIER     "ERROR: Invalid Carrier Number. Try Again."

#define ERR_INVALID_THRESHOLD    "ERROR: Invalid Threshold. Try Again."
#define ERR_INVALID_C_D_ID       "ERROR: Invalid Customer District Id. Try Again."
#define ERR_INVALID_C_W_ID       "ERROR: Invalid Customer Warehouse Id. Try Again."
#define ERR_TERMINAL_FULL        "ERROR: Terminal can not support user. Terminal full."
#define ERR_C_ID_OR_CLAST_ONLY   "ERROR: Either customer id or customer last name can be specified."

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

#define ERR_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
/////////////////////////////////////////////////////////////////
#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
////////////////////////////////////////////////////////////////

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_ords;
    struct out_ordstat_struct out_ords;
};

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 outdeadlocks;
};

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='\0';
}

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

inline void calcOutDateTime(const INT64b value,datetime *timestamp)
{

```

```

// fixed days in each month (FEB 29 is special case)
static int daysInMonth[12]= {31,28,31,30,31,30,31,31,30,31,30,31};

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

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

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

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

int daysUsed = (years * DAYS_IN_YEAR) + leaps;

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

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

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

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

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

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

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

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

    boolnegativeFlag = 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 negitive 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 buf[20];
    int i,j,decimalFound,digitsAfterDecimal=0;

    int decimal=0;

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

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

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

    return amount;
}

//
// copyOutMoney64
//
// Title : Copy money data out of class array
// Parameters: char * - buffer to copy string 64 bit money into
//             INT64b - value
//             unsigned len - max number of bytes to copy
// Return Value: int - Length of copy
// Comments :
//

inline void copyOutMoney64(char *buffer,INT64b value,unsigned int len =
DEFAULT_MONEY64_LEN)
{
    unsigned int index = len;

    int places = 0;

    bool negativeFlag = false;
    bool moneyFlag = true;

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

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

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

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

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

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

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

    // handle negative value
    if(value < 0)
    {
        negitiveFlag = 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 negitive indicator
                    if(negitiveFlag)
                    {
                        negitiveFlag = false;
                        buffer[--index] = NEGITIVE_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); \
    \
    errorStream << debugFileName(__FILE_) \
    << "|" << __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  :
**
*****
*
*/

#ifndef __tpccISAPI_hpp__
#define __tpccISAPI_hpp__

#include <windows.h>
#include <httplib.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
// COM interface
struct COM_HANDLE
{
    Itpcc_com *comHandle;
    char *txnBuffer;
    int size;
};

// Terminal struct
// Terminal struct
// Terminal struct
```

```
// TXN handle
// Terminal struct
// Terminal struct
struct TXN_HANDLE
{
    charhtmlPage[MAX_HTML_PAGE_LEN];
    charhtmlHeader[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 DLVYQUEUEUEDATA
{
    int warehouse;
    short in_s_0_CARRIER_ID;
    struct _timeb enqueueTime;
};

// Definitions
// 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 //One hour in
milli seconds

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

// Function Prototypes
// Function Prototypes
// 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 cmdIDStart);
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 consituant parts
////////////////////////////////////////////////////////////////

#include "htmlPhraser.h"

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

htmlPhraser::htmlPhraser(char *queryString)
{
    // initalize query values
    iCustomerIdFlag = iCarrierNumFlag = iStockThresholdFlag = false;

    // this initalizes 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 characeter of query string
    while(*queryString)
    {
        // check for special case characters

```

```

if(queryChar)
{
    // a percentage sign would indicate a token
    if(*queryString != '%')
    {
        // a plus sign repersents a space
        if(*queryString == '+')
        {
            queryChar = ' ';
            *queryString++;
        }
        else queryChar = *queryString++;
    }
    else queryChar = convertQueryToken(&queryString);
}
else queryChar = '&';

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

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

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

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

    // finished processing for query reference
    continue;
}

// we have a query reference but need to wait until we see '='
// before accepting value

if(valueIndex == -1)
{
    // we are waiting for '='
    if(queryChar == '=')
    {
        valueIndex = 0;

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

    // finishes looging for '='
    continue;
}

// add each character to the query value
if(queryIndex > -1 && valueIndex > -1)

```

```

{
  // we are processing a query value
  if(valueIndex < MAX_FIELD_LEN)
  {
    // we have not exceeded max line len
    iQueryValues[queryIndex][valueIndex++] = queryChar;
  }
  continue;
}
}
return;
}

```

```

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

```

```

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

```

```

// should not get here
return COMMAND_LOGIN;
}

```

```

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

```

```

int validate(int txnType)
{
  return 0;
}

```

```

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

```

```

char htmlPhraser::convertQueryToken(char **queryString)

```

```

{
  char queryChar = NULL;

```

```

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

```

```

  // look at first character
  switch(**queryString)
  {

```

```

    case '2':
      {
        // what follows?
        (*queryString)++;

```

```

        switch(**queryString)

```

```

        {
          case '1':
            queryChar = '!';
            break;

```

```

          case '3':
            queryChar = '#';
            break;

```

```

          case '4':
            queryChar = '$';
            break;

```

```

          case '5':
            queryChar = '%';
            break;

```

```

          case '6':
            queryChar = '&';
            break;

```

```

          case '8':
            queryChar = '(';
            break;

```

```

          case '9':
            queryChar = ')';
            break;

```

```

          case 'B':
            queryChar = '+';
            break;

```

```

          case 'C':
            queryChar = ',';
            break;

```

```

          case 'F':
            queryChar = '/';
            break;

```

```

          case ' ':
            queryChar = ' ';
            break;

```

```

        }
      }

```

```

    break;
    case '3':

```

```

    {
      // what follows?
      (*queryString)++;

```

```

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

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

switch(**queryString)
{
case '0':
    queryChar = '@';
    break;
case ' ':
    queryChar = ' ';
    break;
}
}

break;
case '5':
{
// what follows?

(*queryString)++;

switch(**queryString)
{
case 'B':
    queryChar = '[';
    break;
case 'D':
    queryChar = ']';
    break;
case 'E':
    queryChar = '^';
    break;
case ' ':
    queryChar = ' ';
    break;
}
}

break;
case '7':
{
// what follows?
(*queryString)++;

switch(**queryString)
{
case 'B':
    queryChar = '{';

```

```

break;
case 'C':
    queryChar = '|';
    break;
case 'D':
    queryChar = '}';
    break;
case 'E':
    queryChar = '~';
    break;
case ' ':
    queryChar = ' ';
    break;
}
}

break;
case '+':
    queryChar = '+';
    break;
}

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

```

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

tpccIsapi/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 DLVYQUEUEUEDATA *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];
            sprintf(response,"ERROR: Init txnHandle function failed.\n");

            size_t htmlPageLen = strlen(response);

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

            return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
        }

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

            size_t htmlPageLen = strlen(response);

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

```

```

        return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
    }
}

try
{
    txnHandle->urlString = (char*)lpECB->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);

// Initailize error log critical section
InitializeCriticalSection(&errorLock);

// 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(dlvvy_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 ispaiLock, 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,IID_Itpcc_c
om,(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.txnHande 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 exception in initTxnHandle, unlocking isapi
lock" <<endl);
}

```



```

        ERRORMSG("Unhandled exception 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 = (DLVYQUEUEUDATA *)
calloc(dlvyQueueLen,sizeof(DLVYQUEUEUDATA));

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

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

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

        return ERR_DLVY_EVENT_INIT_FAILED;
    }

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

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

    //index into next available slot in dlvy 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,REGISTRY_SUB_KEY
,0,KEY_READ,&registryKey) != ERROR_SUCCESS)
        return ERR_UNABLE_TO_OPEN_REG;

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

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

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

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

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

    //get the num of users

```

```

    regValueSize = sizeof(regValue);
    if(RegQueryValueEx(registryKey,NUM_USERS,0,&regType,(BYTE
*)&regValue,&regValueSize) == ERROR_SUCCESS)
        numUsers = regValue;
    else
        numUsers = DEFAULT_NUM_USERS;

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

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

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

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

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

    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,"dlvy Queue Length:%d <BR> num dlvy threads:%d
<BR> dlvy queue free slots:%d <BR> isapi queue index:%d <BR> thread
queue index:%d <BR> </BODY></HTML>\n",
dlvyQueueLen,
dlvyThreads,
dlvyBufferFreeSlots,
dlvyBufferSlotIndex,
dlvyBufferThreadIndex);

```

```

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>"
//
" 1 2 3 4 5 6 7 8
9\r\n"
//
"12345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890\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>"
//
" 1 2 3 4 5 6 7 8
9\r\n"
//
"12345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890\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 :
**           HTML neworder page entry point
** Parameters :
**           htmlPhraser* command block
**           TXN_HANDLE* txn handle struct
** Returns   :
**           int - return code
** 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, commandBlock, txnHa
ndle);
        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, commandBlock, txnHa
ndle);
        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_OL_CNT].s_OL_SUPPLY_W_ID = atoi(commandBlock-
>get_ITEM_SUPP_W(itemIndex))) == 0)
            {
                doNewOrderErrorPage(html, ERR_INVALID_SUPPLY_W_ID, commandB
lock, 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,commandBloc
k,txnHandle);
            return OK;
        }
        else
            itemComplete++;
    }
    //missing previous value of item supp warehouse, flag error
    else
    {
        doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandB
lock,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,commandBloc
k,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,commandBlock
.txnHandle);
            return OK;
        }
        else
            itemComplete++;
    }
    //missing previous value of item number
    else if( (itemComplete ==1) )
    {
        doNewOrderErrorPage(html,ERR_INVALID_ITEM_NUM,commandBloc
k,txnHandle);
        return OK;
    }
    //missing 1st value of supp warehouse
    else
    {
        doNewOrderErrorPage(html,ERR_INVALID_SUPPLY_W_ID,commandB
lock,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,commandBloc
k,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"
"
12345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890\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,ittoa(nord->in_nord.s_W_ID,buffer,10),6,1);

appendText(&html,"District: ");
appendText(&html,ittoa(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,ittoa(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,ittoa(nord->out_nord.s_O_ID,buffer,10),10,1);

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

if(rc != INVALID_STATUS)
    appendText(&html,ittoa(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"
//
"12345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890\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++)
    {
        appendText(&html,ittoa(nord-
>in_nord.in_item[itemCount].s_OL_SUPPLY_W_ID,buffer,10),8,1);
        appendText(&html,ittoa(nord-
>in_nord.in_item[itemCount].s_OL_I_ID,buffer,10),10,1);
        appendText(&html,nord-
>out_nord.item[itemCount].s_I_NAME,DEFAULT_STRING_LEN+1,1);
        appendText(&html,ittoa(nord-
>in_nord.in_item[itemCount].s_OL_QUANTITY,buffer,10),5,1);
        appendText(&html,ittoa(nord-
>out_nord.item[itemCount].s_S_QUANTITY,buffer,10),7,1);
        html+=sprintf(html,"%c $%-7.2lf $%-7.2lf <BR> ",nord-
>out_nord.item[itemCount].s_brand_generic,
nord-
>out_nord.item[itemCount].s_I_PRICE/100.0,
nord-
>out_nord.item[itemCount].s_OL_AMOUNT/100.0);
    }
    //display blank line for remaining empty items in the order
    for(int lineBreaks=0;lineBreaks < (NORD_ITEMS-nord-
>out_nord.s_O_OL_CNT);lineBreaks++)
        appendText(&html," <BR>");
}
else
    appendText(&html," <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>");

appendText(&html,"\r\n <BR> ");

html+=displayStatus(html,rc);
if(rc == OK)
    html+=sprintf(html," Total: $%.2lf",nord-
>out_nord.s_total_amount/100.0);
else
    appendText(&html," Total: <BR>");

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

    DEBUGMSG("nord html page complete. returning to calling function" <<
endl);

return OK;
}
/*

```

```

*****
*
** Name      : doNewOrderErrorPage
** Description :
**          HTML neworder page entry point
** Parameters :
**          char * html result page
**          char * error message
**          htmlPhraser* command block
**          TXN_HANDLE* txn handle struct
** Returns   :
**          int - return code
** Comments  :
*****
*/

```

```

int doNewOrderErrorPage(char *htmlPage,char *message,htmlPhraser
*commandBlock,TXN_HANDLE *txnHandle)

```

```

{
    char *html=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"
        "Submit Transaction <INPUT TYPE=\\"submit\\"" NAME=\\""
        CMD_TXN_ID
        "\" VALUE=\\""
        CMD_NORD
        "\">");

    //append the hidden warehouse and district fields
    html+=appendHiddenFields(html,txnHandle);

    //int buffer for warehouse
    char buffer[15];
    /*appendText(&html,"<PRE>      1      2      3      4      5      6
7      8      9\r\n"

"123456789012345678901234567890123456789012345678901234567890123456789012
3456789012345678901234567890\r\n"
    "Warehouse: ");*/
    appendText(&html,"<PRE>Warehouse: ");
    appendText(&html,ittoa(txnHandle->w_id,buffer,10),7,1);
    appendText(&html,"District: <INPUT NAME=\\""
        CMD_D_ID
        "\" SIZE=1>          Date:<BR>"
        "Customer <INPUT NAME=\\""
        CMD_C_ID
        "\" SIZE=6> Name:          Credit:  %Disc.:<BR>"
        "Order Number:      Number of Lines:      W_tax:
D_tax:<BR><BR>"
        //"      1      2      3      4      5      6      7      8
9\r\n"

//"12345678901234567890123456789012345678901234567890123456789012345678901
23456789012345678901234567890\r\n"
        " Supp_W Item_Num Item_Name          Qty Stock B/G
Price Amount <BR> ");

    //append the 15 items commands
    html+=appendItems(html,NORD_ITEMS,ITEM_START);
    appendText(&html,message);

    //seal up html page
    appendText(&html,"</PRE></BODY></HTML>");
}

```

```

return OK;
}

/*
*****
*
** Name      : doPaymentForm
** Description :
**          HTML payment page entry point
** Parameters :
**          htmlPhraser* command block
**          TXN_HANDLE* txn handle struct
** Returns   :
**          int - return code
** Comments  :
*****
*/

```

```

int doPaymentForm(htmlPhraser *commandBlock,TXN_HANDLE
*txnHandle)

```

```

{
    char *html=txnHandle->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,ittoa(txnHandle->w_id,buffer,10));

    appendSpaces(&html,10);
    appendText(&html,"District: <INPUT NAME=\\""
        CMD_D_ID
        "\" SIZE=1>\r\n<BR>"
        "<BR><BR><BR>"
        "Customer: "
        "<INPUT NAME=\\""
        CMD_C_ID
        "\" SIZE=5>"
        " "
        "Cust-Warehouse: "
        "<INPUT NAME=\\""
        CMD_C_W_ID
        "\" SIZE=5>"
        " "
        "Cust-District: "
        "<INPUT NAME=\\""
        CMD_C_D_ID
        "\" SIZE=1><BR>"
        "Name: <INPUT NAME=\\""
        CMD_C_NAME
        "\" SIZE=20>");
    appendText(&html,"          Since: <BR>"
        "          "
        "          Credit: <BR>"
        "          "

```

```

"                %Disc: <BR>"
"Amount Paid:    "
"<INPUT NAME=|"
CMD_AMT_PAID
"\" SIZE=10>"
"                "
"New Cust-Balance:<BR>"
"Credit Limit:<BR> <BR>Cust-Data:<BR> <BR> <BR> <BR>
</PRE>");
return OK;
}

/*
*****
** Name      : doPaymentResults
** Description :
**          HTML neworder page entry point
** Parameters :
**          htmlPhraser* command block
**          TXN_HANDLE* txn handle struct
** Returns   :
**          int - return code
** Comments  :
**
*****
*/

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,commandBlock,txnHan
dle);
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_CLAST,command
Block,txnHandle);
return OK;
}
else
strcpy(pymt->in_paym.s_C_LAST,commandBlock->get_C_NAME());
}
else
{
//make sure that the user only inserted just c_id
if(*(commandBlock->get_C_NAME()) != NULL)
{

```

```

doPaymentErrorPage(html,ERR_C_ID_OR_CLAST_ONLY,commandBloc
k,txnHandle);
return OK;
}
}

//get customer warehouse id field
if( (pymt->in_paym.s_C_W_ID = atoi(commandBlock->get_C_W_ID()))
== 0)
{
doPaymentErrorPage(html,ERR_INVALID_C_W_ID,commandBlock,txnH
andle);
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,commandBlock,txnH
andle);
return OK;
}

if(!copyInMoney64(commandBlock->get_AMT_PAID(),&pymt-
>in_paym.s_H_AMOUNT))
{
doPaymentErrorPage(html,ERR_INVALID_PAYMENT_AMOUNT,comm
andBlock,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></CENTE
R>");

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 = (pymt_wrapper *)txnHandle->comInterface.txnBuffer;
//get return code
int rc = pymt->out_paym.s_transtatus;
if( rc != OK)
{
    html+=displayStatus(html,rc);
    appendText(&html,"</PRE></BODY></HTML>");
    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>\r\n");
// appendText(&html," 1 2 3 4 5 6 7
8<BR>");
//
appendText(&html,"12345678901234567890123456789012345678901234
567890123456789012345678901234567890<BR>");

//start creating result body
appendText(&html, "<BR><PRE>\r\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: ");
copyOutDate(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 :
**      append payment error body
** Parameters :
**      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
        "\">"
        "<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></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,commandBlock,txnH
andle);
        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,comma
ndBlock,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_ONLY,command
Block,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
exception.</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>\r\nWarehouse: ");
    char buffer[50];

    appendText(&html,ittoa(ords->in_ords.s_W_ID,buffer,10),6+1,1);
    appendText(&html,"District: ");
    appendText(&html,ittoa(ords->in_ords.s_D_ID,buffer,10));
    appendText(&html,"<BR>"
"Customer: ");
    //get customer id
    appendText(&html,ittoa(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,"\r\nCust-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,ittoa(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,ittoa(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,ittoa(ords-
>out_ords.item[itemCount].s_OL_SUPPLY_W_ID,buffer,10),11,1);

        //get item num
        appendText(&html,ittoa(ords-
>out_ords.item[itemCount].s_OL_I_ID,buffer,10),11,1);

        //get item oty
        appendText(&html,ittoa(ords-
>out_ords.item[itemCount].s_OL_QUANTITY,buffer,10),6,1);

        //get item dollar amount
        html+=sprintf(html,"% .14lf",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></PRE></BODY><FORM ACTION=\"
\"<BODY><FORM ACTION=\"
APP_NAME
\"\" METHOD=\"GET\"></PRE>
\"<CENTER><H3>Please Fill In Order Status
Form.</H3></CENTER> <BR></PRE>
\"Submit Transaction <INPUT TYPE=\"submit\" NAME=\"
CMD_TXN_ID
\"\" VALUE=\"
CMD_ORDS
\"\">
\"<BR> \");
    html+=appendHiddenFields(html,txnHandle);

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

    appendText(&html,"          District: <INPUT NAME=\"
CMD_D_ID
\"\" SIZE=1></PRE><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></PRE>
\"<BODY><FORM ACTION=\"
APP_NAME
\"\" METHOD=\"GET\"></PRE>
\"<CENTER><H3>Delivery.</H3></CENTER></PRE>
\"Submit Transaction <INPUT TYPE=\"submit\" NAME=\"
CMD_TXN_ID
\"\" VALUE=\"
CMD_DLVY
\"\">");
    html+=appendHiddenFields(html,txnHandle);

    appendText(&html,"<BR> <PRE>
\"Warehouse: \");
    char buffer[10];
    appendText(&html,ittoa(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_dlvvy.s_W_ID = txnHandle->w_id;

//set the carrier id from command blk
if( (dlvy.in_dlvvy.s_O_CARRIER_ID = atoi(commandBlock-
>get_CARRIER_NUM())) == 0)
{
    doDeliveryErrorPage(html,ERR_INVALID_CARRIER,commandBlock,txn
Handle);
    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_dlvvy.s_W_ID,dlvy.in_dlvvy.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_dlvvy.s_W_ID,buffer,10));
appendText(&html,"<BR> <BR>Carrier Number: ");

//get carrier_id from wrapper
appendText(&html,itoa(dlvy.in_dlvvy.s_O_CARRIER_ID,buffer,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
**           htmlPhrasercommand 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_DLVY
\">");
    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 :
**           htmlPhrasercommand 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,itoa(txnHandle->w_id,buffer,10),6+1,1);
appendText(&html,"District: ");

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

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

return OK;
}

/*
*****
** Name      : doStockResults
** Description :
**           HTML stock page entry point
** Parameters :
**           htmlPhraser* command block
**           TXN_HANDLE* 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,commandBlock,tx
nHandle);
        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 exception to
occur.</PRE></BODY></HTML>");
        ERRORMSG("ERROR : Com Stock call caused exception 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,itoa(stok->in_stok.s_W_ID,buffer,10),6+1,1);

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

```

```

appendText(&html," <BR> <BR>"
           "Stock Level Threshold: ");
appendText(&html,ittoa(stok->in_stok.s_threshold,buffer,10));

appendText(&html," <BR> <BR>"
           "Low Stock: ");
appendText(&html,ittoa(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,ittoa(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 appended
**                  to the html page
** Comments   :
**
*****
*/
int appendHiddenFields(char *htmlPage,TXN_HANDLE *txnHandle)
{
    char *html = htmlPage;
    char buffer[15];

    appendText(&html,"<INPUT TYPE=\"hidden\" NAME=\""
        CMD_TERM_ID
        "\" VALUE=\""");
    appendText(&html,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 appended
**      to the html page
** Comments   :
**
*****
*/
int appendButtons(char *htmlPage)
{
    char *html = htmlPage;

    appendText(&html,"<INPUT TYPE=\"submit\" NAME=\""
        CMD_TXN_ID
        "\" VALUE=\""
        CMD_NORD
        "\">\r\n"
        "<INPUT TYPE=\"submit\" NAME=\""
        CMD_TXN_ID
        "\" VALUE=\""
        CMD_PYMT
        "\">\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 :
**      appends items to new order and order status page
** Parameters :
**      *htmlPage      html result page
**      short          items to append
**      short          item CMD id start
**
** Returns    :
**      amount of characters the function appended
**      to the html page
** Comments   :
**
*****
*/
int appendItems(char *htmlPage,short itemCount,short cmdIDStart)
{
    char *html = htmlPage;
    char numBuffer[MAX_INT_BUFFER];

    for(int item=0;item < itemCount;item++)
    {
        appendText(&html,"<BR> <INPUT NAME=\""");
        appendText(&html,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 initDly 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,dlvyThreadID);
    dlvyLog = fopen(logFileName,"w");
    if(!dlvyLog)
    {
        ERRORMSG("dlvyThread " << GetCurrentThreadId() << " unable to
open dlvy log "
        << dlvyLogPath << "\\del_" << dlvyThreadID << endl);
        DEBUGMSG("dlvyThread " << GetCurrentThreadId() << " unable
toopen 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 array to store event to
wait on

    struct DLVYQUEUEDATA  dlvyQueueData;    //dlvy queue struct to
store queued txn
    struct dlvy_wrapper  dlvyTxn;         //dlvy wrapper of db2 structs

    struct _timeb        endQueueTime;     //time stamp to queue removal
time

```

```

    struct _timeb        endProcessTime;    //time stamp for end process
time

    char orderIDs[MAX_STRING_LEN] = {NULL}; //string to store oids
for each district
    int    bytesWritten = 0;
    int    dlvyCount = 0;

    DEBUGMSG("dlvyThread entering work loop" << endl);

    //successful, while true
    while(true)
    {
        try
        {
            DEBUGMSG("dlvyThread initializing wait handles" << endl);

            //wait for both program exit AND if there is work to do
            workerHandles[0] = dlvyThreadDone;
            workerHandles[1] = dlvyThreadSemaphore;

            DEBUGMSG("dlvyThread going to sleep waiting for wrk" << endl);

            rc =
            WaitForMultipleObjects(2,&workerHandles[0],FALSE,INFINITE);

            DEBUGMSG("dlvyThread awake, checking wake condition" << endl);

            if(rc == WAIT_OBJECT_0)
                break;
            else if(rc == (WAIT_OBJECT_0+1) )
            {
                DEBUGMSG("dlvyThread awake, wake condition of
dlvyThreadSemaphore" << endl);
            }

            DEBUGMSG("dlvyThread trying to enter critical section" << endl);

            EnterCriticalSection(&dlvyQueueLock);

            DEBUGMSG("dlvyThread entered critical section" << endl);

            //remove queued dlvy txn
            dlvyQueueData.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_O_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_O_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);
        if( dlvyBufferSlotIndex == (dlvyBufferThreadIndex-1))
        {
            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_DLVEY_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);
htmlPhraser commandBlock(txnHandle->urlString);
DEBUGMSG("Query string parsed. command:"<<
commandBlock.getCommandId() <<" user's terminal id:" <<
commandBlock.get_TERM_ID() << endl);

int terminalID = atoi(commandBlock.get_TERM_ID());

```

```

int commandID = commandBlock.getCommandId();

DEBUGMSG("User sent in a 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()](&commandBlock,txn
Handle);
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=-qlag=i:i -qlanglvl=ansi -qpluscmt -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=CK040116

# 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.
**
*****
*/

#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

```

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 81700
#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 <unistd.h>
#include <string.h>
#include <time.h>

#define daricall

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

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

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

```



```

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

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

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

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

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

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

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

#endif // __TPCCAPP_H

```

include/tpccdbg.h

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 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
 *
 */

#endif __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 =tpccmisc$(OBJEXT) tpccdbg$(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: connect
  db2 bind tpccctx.bnd $(BND_OPTS)

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

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

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

```

```

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

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

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

```

Src.Common/tpccctx.sqc

```

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

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

#include <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", __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", __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.
*****/
/

/*
 * tcdbg.c - Debugging Routines
 */

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

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

#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128

voiddel_print();

```

```

voidnew_print();
voidord_print();
voidpay_print();
voidstk_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, int ptat, SQL_STRUCTURE sqlca
*psqlca)
{
    FILE *err_fn = 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(terr_fn, "stk.err.out");
    strcpy(tranName, "STOCK_LVL");
    break;

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

default:
    return;
}

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

terr_fp = fopen(terr_fn, "a+");

fprintf(terr_fp, "-----\n");
fprintf(terr_fp, "Transaction: %s at point %2d (%s)\n", tranName, ptat, msg);
fprintf(terr_fp, "SQLCODE %d ", psqlca->sqlcode);
fprintf(terr_fp, "PID %d ", getpid());
fprintf(terr_fp, "TIME %s\n", timeStamp);
fprintf(terr_fp, "-----\n");
fprintf(terr_fp, "%s", terrStr);
fprintf(terr_fp, "-----\n");

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

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

```



```

{
char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

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

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

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

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

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

fprintf(debug_fp, "\n=====
=====n");

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

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

```

```

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

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

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

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

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

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

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

fprintf(debug_fp, "\n=====
=====n");

fprintf(debug_fp, "in_payment_struct {\n");
fprintf(debug_fp, "\ts_H_AMOUNT = %lld (%lX)\n",
in_payment->s_H_AMOUNT, in_payment->s_H_AMOUNT);
fprintf(debug_fp, "\ts_C_ID = %d (%X)\n",
in_payment->s_C_ID, in_payment->s_C_ID);
fprintf(debug_fp, "\ts_W_ID = %d (%X)\n",
in_payment->s_W_ID, in_payment->s_W_ID);

```

```

fprintf(debug_fp, "\ts_D_ID      = %d (%X)\n",
        in_payment->s_D_ID, in_payment->s_D_ID);
fprintf(debug_fp, "\ts_C_D_ID      = %d (%X)\n",
        in_payment->s_C_D_ID, in_payment->s_C_D_ID);
fprintf(debug_fp, "\ts_C_W_ID      = %d (%X)\n",
        in_payment->s_C_W_ID, in_payment->s_C_W_ID);
fprintf(debug_fp, "\ts_C_LAST      = %s\n",
        in_payment->s_C_LAST);
fprintf(debug_fp, "\ts_H_DATE      = %lld (%lX)\n",
        in_payment->s_H_DATE_time, in_payment->s_H_DATE_time);
fprintf(debug_fp, "\n}\n");

fprintf(debug_fp, "out_payment_struct {\n");
fprintf(debug_fp, "\ts_H_DATE      = %lld (%lX)\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");

```

```

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");

    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");
    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)
{
    return(time(NULL));
}

/* Current time in SECONDS, precision MILLISECONDS */
double current_time_ms(void)
{
    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

LDFLAGS=$(LDFLAGS_STORP) $(LDFLAGS_LIB)

#
#####
#####
# File Collections
#
#####
#####
STORED_PROCS = new ord del

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

EXE = news ords dels

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

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

clean: connect uncatlog 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: uncatlog
    - 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

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

```



```

- db2 -tvf uncat-proc.ddl +o -z uncat-proc.out

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

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

connect:
- db2 connect to $(TPCC_DBNAME)

disconnect:
- db2 connect reset
- db2 terminate

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

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

#
#####
#####
# 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

# 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
        , OL_NUMBER 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 )
            WHEN SMALLINT( 3 )
            WHEN SMALLINT( 4 )
            WHEN SMALLINT( 5 )
            WHEN SMALLINT( 6 )
            WHEN SMALLINT( 7 )
            WHEN SMALLINT( 8 )
            WHEN SMALLINT( 9 )
            WHEN SMALLINT( 10 )
        THEN S_DIST_02
        THEN S_DIST_03
        THEN S_DIST_04
        THEN S_DIST_05
        THEN S_DIST_06
        THEN S_DIST_07
        THEN S_DIST_08
        THEN S_DIST_09
        THEN S_DIST_10
        END
        WHERE S_I_ID = NEW_OL_ALL.I_ID
        AND S_W_ID = NEW_OL_ALL.SUPP_W_ID
    ) AS U
;
RETURN VALUES( VAR.I_PRICE

```

```

        , VAR.I_NAME
        , VAR.I_DATA
        , VAR.OL_DIST_INFO
        , VAR.S_DATA
        , VAR.S_QUANTITY
    )
;
END
$
CREATE FUNCTION NEW_OL_LOCAL( I_ID INT
    , I_QTY SMALLINT
    , W_ID INT
    , O_ID INT
    , D_ID SMALLINT
    , OL_NUMBER 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
);
-- We do not add an if since the complexity is bigger than drawing the blanks
-- IF I_ID IS NULL THEN
SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )
= ( SELECT OL_DIST_INFO
    , S_DATA
    , S_QUANTITY
FROM NEW TABLE ( UPDATE STOCK
INCLUDE ( OL_DIST_INFO CHAR( 24 ) )
SET S_QUANTITY = CASE WHEN S_QUANTITY -
NEW_OL_LOCAL.I_QTY >= 10
THEN S_QUANTITY -
NEW_OL_LOCAL.I_QTY

```

```

ELSE S_QUANTITY -
NEW_OL_LOCAL.I_QTY + 91
END
, S_ORDER_CNT = S_ORDER_CNT +
SMALLINT( 1 )
, S_YTD = S_YTD +
NEW_OL_LOCAL.I_QTY
, OL_DIST_INFO = CASE D_ID WHEN
SMALLINT( 1 ) THEN S_DIST_01
WHEN SMALLINT( 2 )
THEN S_DIST_02
WHEN SMALLINT( 3 )
THEN S_DIST_03
WHEN SMALLINT( 4 )
THEN S_DIST_04
WHEN SMALLINT( 5 )
THEN S_DIST_05
WHEN SMALLINT( 6 )
THEN S_DIST_06
WHEN SMALLINT( 7 )
THEN S_DIST_07
WHEN SMALLINT( 8 )
THEN S_DIST_08
WHEN SMALLINT( 9 )
THEN S_DIST_09
WHEN SMALLINT( 10 )
THEN S_DIST_10
END
WHERE S_I_ID = NEW_OL_LOCAL.I_ID
AND S_W_ID = NEW_OL_LOCAL.W_ID
) AS U
)
;
RETURN VALUES( VAR.I_PRICE
    , VAR.I_NAME
    , VAR.I_DATA
    , VAR.OL_DIST_INFO
    , VAR.S_DATA
    , VAR.S_QUANTITY
)
;
END
$
CREATE FUNCTION NEW_WH ( O_ID INTEGER
    , W_ID INTEGER
    , D_ID SMALLINT
    , C_ID INTEGER
    , O_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/sqllib/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/sqllib/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/sqllib/function/dels!dels'
not fenced;

```

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.dd

```

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;

```


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
, OL_NUMBER
)
) 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
                    , OL_NUMBER
                    )
) 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
        , OL_NUMBER
    )
) 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_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
, OL_NUMBER
)
) 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_DATA, S_QUANTITY

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

FROM Table( VALUES

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

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

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

FROM Table( VALUES

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

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

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

```

```

        ) 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

FROM Table( VALUES

```

```

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

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

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

) 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
, OL_NUMBER
)
) 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
, 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 )

```

```

        ) 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
        , OL_NUMBER
        )
        ) 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_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
, OL_NUMBER
)
) 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
, OL_NUMBER
)
) 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 )
( 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
, OL_NUMBER
)
) 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
                    ,OL_NUMBER
                )
            ) 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
                    ,OL_NUMBER
                )
            ) 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
    )

    FROM ( SELECT :next_o_id as O_ID
            , :w_id AS W_ID

```

```

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
, 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 )
( 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
, OL_NUMBER
)
) 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
, OL_NUMBER
)
) 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
, OL_NUMBER
)
) 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
, OL_NUMBER
) 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
, OL_NUMBER
) 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
, OL_NUMBER
)
) 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 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
        , OL_NUMBER
        )
) 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
        , OL_NUMBER
        )
) 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
, 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 )

) AS X ( OL_NUMBER , I_ID ,
I_QTY
)

) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
, OL_NUMBER
)
) 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
, 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 )

) AS X ( OL_NUMBER , I_ID ,
I_QTY
)

) AS ITEMLIST

, TABLE( NEW_OL_LOCAL( I_ID
, I_QTY
, W_ID
, O_ID
, D_ID
, OL_NUMBER
)
) 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

```

```

        ) AS X ( OL_NUMBER , I_ID ,
I_QTY      )
        ) AS ITEMLIST
        , TABLE( NEW_OL_LOCAL( I_ID
        , I_QTY
        , W_ID
        , O_ID
        , D_ID
        , OL_NUMBER
        )
        ) 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
        , 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 )
            , ( 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
        , OL_NUMBER
        )
        ) 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_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
, OL_NUMBER
)
) 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
        , OL_NUMBER
    )
    ) 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
, OL_NUMBER
)
) 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
, OL_NUMBER
)
) 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
        , OL_NUMBER
        )
        ) 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", __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", __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", __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", __LINE__, &sqlca) ;
                goto ferror;
            }
        }
    }
    else

```

```

{
    newword->s_transtatus = INVALID_ITEM ;

    EXEC SQL ROLLBACK WORK ;

    if ( sqlca.sqlcode != 0 )
    {
        newword->s_transtatus = FATAL_SQLERROR;

        sqlerror(NEWWORD_SQL, "ROLLBACK FAILED (INVALID
ITEM)", __LINE__, &sqlca);
        // no point in ferror
    }
}
else
{
    DLCHK( retry_tran );

    sqlerror( NEWWORD_SQL, "NEW_WH", __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( newword, in_newword, "SP prior to return");
#endif

return ( storedProcRc );

sql_error:

{
    char tempstr[ 4096 ] ;

    DLCHK( retry_tran );

    sprintf( tempstr,
"inputItemCount=%d, :next_o_id=%d, :d_id=%d, :w_id=%d",
inputItemCount, next_o_id, d_id, w_id );
    sqlerror( NEWWORD_SQL, tempstr, __LINE__, &sqlca );
}

ferror:

newword->s_transtatus = FATAL_SQLERROR;

EXEC SQL ROLLBACK WORK;

if ( sqlca.sqlcode != 0 )
{
    sqlerror( NEWWORD_SQL, "ROLLBACK FAILED", __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, /*60-69*/
    8,4,8,3,8,8,0,8,2,7, /*70-79*/
    8,8,6,8,8,8,8,8,8, /*80-89*/
    8,8,8,8,8,8,8,8,8, /*90-99*/
    8,8,8,8,8,8,8,8,8, /*100-109*/
    8,8,8,8,8,8,8,8,8, /*110-119*/
    8,8,8,8,8,8,8,8,8, /*120-129*/
    8,8,8,8,8,8,8,8,8, /*130-139*/
    8,8,8,8,8,8,8,8,8, /*140-149*/
    8,8,8,8,8,8,8,8,8, /*150-159*/
    8,8,8,8,8,8,8,8,8, /*160-169*/
    8,8,8,8,8,8,8,8,8, /*170-179*/
    8,8,8,8,8,8,8,8,8, /*180-189*/
    8,8,8,8,8,8,8,8,8, /*190-199*/
    8,8,8,8,8,8,8,8,8, /*200-209*/
    8,8,8,8,8,8,8,8,8, /*210-219*/
    8,8,8,8,8,8,8,8,8, /*220-229*/
    8,8,8,8,8,8,8,8,8, /*230-239*/
    8,8,8,8,8,8,8,8,8, /*240-249*/
    8,8,8,8,8}; /*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;

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

    int storedProcRc ;
    int itemArrayIndex = 0 ;

    #define w_id in_ordstat->s_W_ID ;
    #define d_id in_ordstat->s_D_ID ;
    #define c_id_input in_ordstat->s_C_ID
    #define o_id ordstat->s_O_ID
    #define o_entry_d ordstat->s_O_ENTRY_D_time
    #define o_carrier_id ordstat->s_O_CARRIER_ID
    #define c_id ordstat->s_C_ID
    #define c_first ordstat->s_C_FIRST
    #define c_middle ordstat->s_C_MIDDLE
    #define c_last ordstat->s_C_LAST
    #define c_balance ordstat->s_C_BALANCE

```

```

EXEC SQL DECLARE read_orderline_cur CURSOR FOR

    SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY,
    OL_AMOUNT, OL_DELIVERY_D

    FROM ORDER_LINE

    WHERE OL_W_ID = :w_id
    AND OL_D_ID = :d_id
    AND OL_O_ID = :o_id

    FOR FETCH ONLY ;

ordstat->deadlocks = -1 ;

#ifdef DEBUGIT
    ord_debug(ordstat, in_ordstat, "SP upon entry");
#endif

retry_tran:

    ordstat->deadlocks ++ ;

    if ( c_id_input == 0 )
    {
        c_last_input.len = strlen( in_ordstat->s_C_LAST ) ;
        memcpy( c_last_input.data , in_ordstat->s_C_LAST , c_last_input.len ) ;

        EXEC SQL

            SELECT O_ID, O_CARRIER_ID, O_ENTRY_D, C_BALANCE,
            C_FIRST, C_MIDDLE, C_ID

        INTO :o_id, :o_carrier_id, :o_entry_d, :c_balance, :c_first, :c_middle, :c_id

            FROM TABLE ( ORD_C_LAST( :w_id
                , :d_id
                , :c_last_input
                )
            ) AS ORD_C_LAST
        ;
    }
    else
    {
        EXEC SQL

            SELECT O_ID, O_CARRIER_ID, O_ENTRY_D, C_BALANCE,
            C_FIRST, C_MIDDLE, C_LAST

        INTO :o_id, :o_carrier_id, :o_entry_d, :c_balance, :c_first, :c_middle, :c_last

            FROM TABLE ( ORD_C_ID( :w_id
                , :d_id
                , :c_id_input
                )
            ) AS ORD_C_ID
        ;
    }

    if ( sqlca.sqlcode != 0 )
    {
        DLCHK( retry_tran );
        sqlerror( ORDSTAT_SQL, "READ CUST and ORDERS", __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",
__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",
__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", __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", __LINE__, &sqlca);
}

goto mexit;
}

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

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

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

    struct sqlca sqlca ;

    int storedProcRc ;

    short district_id ;
    sqlint32 customer_id ;

    EXEC SQL BEGIN DECLARE SECTION;

    // input

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

    // output

    short no_o_id_indicator = 0 ;
    sqlint32 no_o_id ;

    EXEC SQL END DECLARE SECTION;

#define d_id      district_id
#define c_id      customer_id

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

    delivery->deadlocks = -1 ;

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

```

```

d_id = 1;
retry_tran:

delivery->deadlocks++;

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

EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

SELECT O_ID

INTO :no_o_id :no_o_id_indicator

FROM TABLE
( DEL( :w_id , :d_id , :o_carrier_id , :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", __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", __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 ) ;
}

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(4) 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,
             EXPLAIN_LEVEL,
             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,
EXTENTS_SIZE    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,
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,
    PREDICATE_TEXT 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_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_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)
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;

--
-- 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);

--
-- 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

```

// comreg.h : Declaration of the CCompReg

#pragma once

#include "resource.h" // main symbols
#include "tpccCom.h"

// CCompReg
class ATL_NO_VTABLE CCompReg :
public CComObjectRootEx<CComSingleThreadModel>,
public CComCoClass<CCompReg, &CLSID_CompReg>,
public IDispatchImpl<IComponentRegistrar, &IID_IComponentRegistrar,
&LIBID_tpccComLib, /*wMajor=*/ 1, /*wMinor=*/ 0>
{
public:

```



```

CCompReg()
{
}

DECLARE_NO_REGISTRY()

BEGIN_COM_MAP(CCompReg)
    COM_INTERFACE_ENTRY(IComponentRegistrar)
    COM_INTERFACE_ENTRY(IDispatch)
END_COM_MAP()

// IComponentRegistrar
public:
    STDMETHOD(Attach)(BSTR bstrPath)
    {
        return S_OK;
    }
    STDMETHOD(RegisterAll)()
    {
        return _AtlComModule.RegisterServer(TRUE);
    }
    STDMETHOD(UnregisterAll)()
    {
        _AtlComModule.UnregisterServer(TRUE);
        return S_OK;
    }
    STDMETHOD(GetComponents)(SAFEARRAY **ppCLSIDs,
SAFEARRAY **ppDescriptions)
    {
        if (ppCLSIDs == NULL || ppDescriptions == NULL )
            return E_POINTER;
        int nComponents = 0;
        for (_ATL_OBJMAP_ENTRY** ppEntry =
_AtlComModule.m_ppAutoObjMapFirst; ppEntry <
_AtlComModule.m_ppAutoObjMapLast; ppEntry++)
        {
            if (*ppEntry != NULL)
            {
                _ATL_OBJMAP_ENTRY* pEntry = *ppEntry;
                if (pEntry->pclsid != NULL)
                {
                    LPCTSTR pszDescription = pEntry->pfnGetObjectDescription();
                    if (pszDescription)
                        nComponents++;
                }
            }
        }
        SAFEARRAYBOUND rgBound[1];
        rgBound[0].lLbound = 0;
        rgBound[0].cElements = nComponents;
        *ppCLSIDs = SafeArrayCreate(VT_BSTR, 1, rgBound);
        if (*ppCLSIDs == NULL )
            return AtlHresultFromLastError();
        *ppDescriptions = SafeArrayCreate(VT_BSTR, 1, rgBound);
        if (*ppDescriptions == NULL )
            return AtlHresultFromLastError();
        LONG i = 0;
        for (_ATL_OBJMAP_ENTRY** ppEntry =
_AtlComModule.m_ppAutoObjMapFirst; ppEntry <
_AtlComModule.m_ppAutoObjMapLast; ppEntry++)
        {
            if (*ppEntry != NULL)
            {
                _ATL_OBJMAP_ENTRY* pEntry = *ppEntry;
                if (pEntry->pclsid != NULL)
                {
                    LPCTSTR pszDescription = pEntry->pfnGetObjectDescription();
                    if (pszDescription)
                    {
                        LPOLESTR pszCLSID;
                        StringFromCLSID(*pEntry->pclsid, &pszCLSID);
                        BSTR pBSTR = OLE2BSTR(pszCLSID);

```

```

        if ( pBSTR == NULL )
        {
            CoTaskMemFree(pszCLSID);
            return E_OUTOFMEMORY;
        }
        HRESULT hResult = SafeArrayPutElement(*ppCLSIDs, &i,
pBSTR);
        CoTaskMemFree(pszCLSID);
        if ( FAILED(hResult) )
            return hResult;
        pBSTR = T2BSTR_EX(pszDescription);
        if( pBSTR == NULL )
        {
            return E_OUTOFMEMORY;
        }
        hResult = SafeArrayPutElement(*ppDescriptions, &i, pBSTR);
        if( FAILED(hResult) )
            return hResult;
        i++;
    }
}

return S_OK;
}
STDMETHOD(RegisterComponent)(BSTR bstrCLSID)
{
    CLSID clsid;
    CLSIDFromString(bstrCLSID, &clsid);
    _AtlComModule.RegisterServer(TRUE, &clsid);
    return S_OK;
}
STDMETHOD(UnregisterComponent)(BSTR bstrCLSID)
{
    CLSID clsid;
    CLSIDFromString(bstrCLSID, &clsid);
    _AtlComModule.UnregisterServer(FALSE, &clsid);
    return S_OK;
}
};

OBJECT_ENTRY_AUTO(CLSID_CompReg, CCompReg)

```

tpccCom/dlldatax.h

```

#pragma once

#ifdef _MERGE_PROXYSTUB

extern "C"
{
    BOOL WINAPI PrxDllMain(HINSTANCE hInstance, DWORD dwReason,
LPVOID lpReserved);
    STDAPI PrxDllCanUnloadNow(void);
    STDAPI PrxDllGetObjectClassObject(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
#ifdef 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 "..\tpccIsapi\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 Itpcc_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_debug.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:
    STDMETHODCALLTYPE(Activate());

```

```

    STDMETHODCALLTYPE(BOOL, CanBePooled());
    STDMETHODCALLTYPE(void, Deactivate());
    CComPtr<IObjectContext> m_spObjectContext;

// Itpcc_com
public:
    STDMETHODCALLTYPE(doStockLevel)(INT *size, UCHAR **buffer);
    STDMETHODCALLTYPE(doNewOrder)(INT* size, UCHAR** buffer);
    STDMETHODCALLTYPE(doPayment)(INT* size, UCHAR** buffer);
    STDMETHODCALLTYPE(doOrderStatus)(INT* size, UCHAR** buffer);
    STDMETHODCALLTYPE(doDBInfo)(void);
    STDMETHODCALLTYPE(doSetComplete)(void);

    int connected;
    int connectHandleInUse;

private:
    //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\api\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)(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_connection;
DISCONNECT_PTR do_disconnect;

```

```

// Ctpcc_com
class ATL_NO_VTABLE Ctpcc_com :
public CComObjectRootEx<CComMultiThreadModel>,
public IObjectControl,
public CComCoClass<Ctpcc_com, &CLSID_tpcc_com>,
public Itpcc_com
{
public:
Ctpcc_com()
{
int rc = ERR;
connected = 0;
connectHandleInUse = 0;

if(debugFlag)
{
if(!debugFileOpen)
{
InitializeCriticalSection(&debugMutex);
//open comLog
char comLogFile[128];

sprintf(comLogFile,"C:\\inetpub\\wwwroot\\tpcc\\comLog_debug.txt");
debugStream.rdbuf()->open(comLogFile,ios_base::in | ios_base::out |
ios_base::app);

debugFileOpen = 1;
}
}

//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 "oaidl.idl";
import "ocidl.idl";
//this is test.

[
    object,
    uuid(a817e7a2-43fa-11d0-9e44-00aa00b6770a),
    dual,
    helpstring("IComponentRegistrar Interface"),
    pointer_default(unique)
]
interface IComponentRegistrar : IDispatch
{
    [id(1)] HRESULT Attach([in] BSTR bstrPath);
    [id(2)] HRESULT RegisterAll();
    [id(3)] HRESULT UnregisterAll();
    [id(4)] HRESULT GetComponents([out] SAFEARRAY(BSTR)*
    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

```

// comreg.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_TPCCCOM,
"{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 DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
{
#ifdef _MERGE_PROXYSTUB
    if (PrxDllGetClassObject(rclsid, riid, ppv) == S_OK)
        return S_OK;
#endif
    return _AtlModule.DllGetClassObject(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_pym.s_W_ID<<" d_id:"<< pymt->in_pym.s_D_ID<<
    " s_transtatus:"<<pymt->out_pym.s_transtatus<<endl);

    do_pymt(pymt,connectHandle);

    DEBUGMSG("Return from do_pymt call using
connectHandle:"<<DEBUGADDRESS(connectHandle)<<" w_id:"<<pymt-
>in_pym.s_W_ID<<" d_id:"<< pymt->in_pym.s_D_ID<<
    " s_transtatus:"<<pymt->out_pym.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 appioate 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"<<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_pynt function address from "<<dbType<<"
dll"<<endl);
if( do_pynt = (PYMT_PTR) GetProcAddress(dbInstance,"do_pynt")
== NULL)
    return ERR_PYMT_ADDRESS_NOT_FOUND;
DEBUGMSG("do_pynt function
address:"<<DEBUGADDRESS(do_pynt)<<endl);

    DEBUGMSG("Getting do_ords function address from "<<dbType<<"
dll"<<endl);
if( do_ords = (ORDS_PTR) GetProcAddress(dbInstance,"do_ords") ==
NULL)
    return ERR_ORDS_ADDRESS_NOT_FOUND;
DEBUGMSG("do_ords function
address:"<<DEBUGADDRESS(do_ords)<<endl);

    DEBUGMSG("Getting do_stok function address from "<<dbType<<"
dll"<<endl);
if( do_stok = (STOK_PTR) GetProcAddress(dbInstance,"do_stok") ==
NULL)
    return ERR_STOK_ADDRESS_NOT_FOUND;
DEBUGMSG("do_stok function
address:"<<DEBUGADDRESS(do_stok)<<endl);

    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,REGISTRY_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/dllldata.c

```

/*****

```

DllData file -- generated by MIDL compiler

DO NOT ALTER THIS FILE

This file is regenerated by MIDL on every IDL file compile.

To completely reconstruct this file, delete it and rerun MIDL on all the IDL files in this DLL, specifying this file for the /dlldata command line option

```

*****/

```

```

#define PROXY_DELEGATION

```

```

#include <rpcproxy.h>

```

```

#ifdef __cplusplus

```

```

extern "C" {
#endif

```

```

EXTERN_PROXY_FILE( tpccCom )

```

```

PROXYFILE_LIST_START

```

```

/* Start of list */
REFERENCE_PROXY_FILE( tpccCom ),
/* End of list */
PROXYFILE_LIST_END

```

```

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

```

```

#ifdef __cplusplus
} /*extern "C" */
#endif

```

```

/* end of generated dlldata file */

```

tpccCom/dlldata.c

```

// wrapper for dlldata.c

```

```

#ifdef _MERGE_PROXYSTUB // merge proxy stub DLL

```

```

#define REGISTER_PROXY_DLL //DllRegisterServer, etc.

```

```

#define _WIN32_WINNT 0x0500 //for Win2000, change it to 0x0400 for
NT4 or Win95 with DCOM
#define USE_STUBLESS_PROXY //defined only with MIDL switch /Oicf

```

```

#pragma comment(lib, "rpcns4.lib")
#pragma comment(lib, "rpcrt4.lib")

```

```

#define ENTRY_PREFIX Prx

```

```

#include "dlldata.c"
#include "tpccCom_p.c"

```

```

#endif // _MERGE_PROXYSTUB

```

tpccCom/tpccCom_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( )

```

```

#ifdef !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

#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,0x44,0x00,0xaa,0x00,0xb6,0x77,0x0a);

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

MIDL_DEFINE_GUID(IID,
LIBID_tpccComLib,0x91F1B8B0,0x89E9,0x457B,0xA2,0x28,0x3E,0x2D,0x6C,0xE3,0xE7,0x52);

MIDL_DEFINE_GUID(CLSID,
CLSID_CompReg,0x90EEDAFF,0xF8D3,0x4711,0x99,0xA9,0x8A,0xC3,0xC0,0xFE,0x5D,0xB9);

MIDL_DEFINE_GUID(CLSID,
CLSID_tpcc_com,0x5F752BF2,0xF739,0x43D4,0x84,0x92,0x44,0xC1,0x95,0x81,0xC0,0xA1);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

```

```

#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,0x10,0x48,0x
60}},{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, */
        0x2, /* 2 */
/* 16 */ 0x8, /* 8 */
        0x5, /* Ext Flags: new corr desc, srv corr check, */
/* 18 */ NdrFcShort( 0x0 ), /* 0 */
/* 20 */ NdrFcShort( 0x1 ), /* 1 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */

        /* Parameter bstrPath */

/* 24 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 26 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 28 */ NdrFcShort( 0x1c ), /* Type Offset=28 */

        /* Return value */

```

```

/* 30 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 32 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 34 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure doSetComplete */

        /* Procedure RegisterAll */

/* 36 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */
/* 38 */ NdrFcLong( 0x0 ), /* 0 */
/* 42 */ NdrFcShort( 0x8 ), /* 8 */
/* 44 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 46 */ NdrFcShort( 0x0 ), /* 0 */
/* 48 */ NdrFcShort( 0x8 ), /* 8 */
/* 50 */ 0x44, /* Oi2 Flags: has return, has ext, */
        0x1, /* 1 */
/* 52 */ 0x8, /* 8 */
        0x1, /* Ext Flags: new corr desc, */
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x0 ), /* 0 */
/* 58 */ NdrFcShort( 0x0 ), /* 0 */

        /* Return value */

        /* Return value */

/* 60 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 62 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 64 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure UnregisterAll */

/* 66 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */
/* 68 */ NdrFcLong( 0x0 ), /* 0 */
/* 72 */ NdrFcShort( 0x9 ), /* 9 */
/* 74 */ NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 76 */ NdrFcShort( 0x0 ), /* 0 */
/* 78 */ NdrFcShort( 0x8 ), /* 8 */
/* 80 */ 0x44, /* Oi2 Flags: has return, has ext, */
        0x1, /* 1 */
/* 82 */ 0x8, /* 8 */
        0x1, /* Ext Flags: new corr desc, */
/* 84 */ NdrFcShort( 0x0 ), /* 0 */
/* 86 */ NdrFcShort( 0x0 ), /* 0 */
/* 88 */ NdrFcShort( 0x0 ), /* 0 */

        /* Return value */

/* 90 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 92 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 94 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure 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 pbstrDescriptions */

/* 126 */NdrFcShort( 0x2113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=8 */
/* 128 */NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 130 */NdrFcShort( 0x41e ), /* Type Offset=1054 */

/* Return value */

/* 132 */NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 134 */NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 136 */0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure RegisterComponent */

/* 138 */0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 140 */NdrFcLong( 0x0 ), /* 0 */
/* 144 */NdrFcShort( 0xb ), /* 11 */
/* 146 */NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 148 */NdrFcShort( 0x0 ), /* 0 */
/* 150 */NdrFcShort( 0x8 ), /* 8 */
/* 152 */0x46, /* Oi2 Flags: clt must size, has return, has ext, */
0x2, /* 2 */
/* 154 */0x8, /* 8 */
0x5, /* Ext Flags: new corr desc, srv corr check, */
/* 156 */NdrFcShort( 0x0 ), /* 0 */
/* 158 */NdrFcShort( 0x1 ), /* 1 */
/* 160 */NdrFcShort( 0x0 ), /* 0 */

/* Parameter bstrCLSID */

/* 162 */NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 164 */NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 166 */NdrFcShort( 0x1c ), /* Type Offset=28 */

/* Return value */

/* 168 */NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 170 */NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 172 */0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure UnregisterComponent */

/* 174 */0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 176 */NdrFcLong( 0x0 ), /* 0 */
/* 180 */NdrFcShort( 0xc ), /* 12 */
/* 182 */NdrFcShort( 0xc ), /* x86 Stack size/offset = 12 */
/* 184 */NdrFcShort( 0x0 ), /* 0 */
/* 186 */NdrFcShort( 0x8 ), /* 8 */
/* 188 */0x46, /* Oi2 Flags: clt must size, has return, has ext, */
0x2, /* 2 */
/* 190 */0x8, /* 8 */
0x5, /* Ext Flags: new corr desc, srv corr check, */
/* 192 */NdrFcShort( 0x0 ), /* 0 */
/* 194 */NdrFcShort( 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( 0xe ), /* 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( 0xffff4 ), /* Offset= -12 (712) */
/* 726 */
0x13, 0x8, /* FC_OP [simple_pointer] */
/* 728 */0x2, /* FC_CHAR */
0x5c, /* FC_PAD */
/* 730 */
0x1a, /* FC_BOGUS_STRUCT */
0x7, /* 7 */
/* 732 */NdrFcShort( 0x20 ), /* 32 */
/* 734 */NdrFcShort( 0x0 ), /* 0 */
/* 736 */NdrFcShort( 0x0 ), /* Offset= 0 (736) */
/* 738 */0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 740 */0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 742 */0x6, /* FC_SHORT */
0x6, /* FC_SHORT */
/* 744 */0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 746 */NdrFcShort( 0xfe30 ), /* Offset= -464 (282) */
/* 748 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 750 */
0x1b, /* FC_CARRAY */

```

```

0x3, /* 3 */
/* 752 */NdrFcShort( 0x4 ), /* 4 */
/* 754 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 756 */NdrFcShort( 0x0 ), /* 0 */
/* 758 */NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 760 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 762 */
0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 764 */NdrFcShort( 0x4 ), /* 4 */
/* 766 */NdrFcShort( 0x0 ), /* 0 */
/* 768 */NdrFcShort( 0x1 ), /* 1 */
/* 770 */NdrFcShort( 0x0 ), /* 0 */
/* 772 */NdrFcShort( 0x0 ), /* 0 */
/* 774 */0x13, 0x0, /* FC_OP */
/* 776 */NdrFcShort( 0xffd2 ), /* Offset= -46 (730) */
/* 778 */
0x5b, /* FC_END */

0x8, /* FC_LONG */
/* 780 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 782 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 784 */NdrFcShort( 0x8 ), /* 8 */
/* 786 */NdrFcShort( 0x0 ), /* 0 */
/* 788 */NdrFcShort( 0x6 ), /* Offset= 6 (794) */
/* 790 */0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 792 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 794 */
0x11, 0x0, /* FC_RP */
/* 796 */NdrFcShort( 0xffd2 ), /* Offset= -46 (750) */
/* 798 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 800 */NdrFcShort( 0x4 ), /* 4 */
/* 802 */0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 804 */NdrFcShort( 0x0 ), /* 0 */
/* 806 */NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 808 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 810 */
0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 812 */NdrFcShort( 0x4 ), /* 4 */
/* 814 */NdrFcShort( 0x0 ), /* 0 */
/* 816 */NdrFcShort( 0x1 ), /* 1 */
/* 818 */NdrFcShort( 0x0 ), /* 0 */
/* 820 */NdrFcShort( 0x0 ), /* 0 */
/* 822 */0x13, 0x0, /* FC_OP */
/* 824 */NdrFcShort( 0xff40 ), /* Offset= -192 (632) */
/* 826 */
0x5b, /* FC_END */

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( 0xff1 ), /* Offset= -15 (846) */
0x5b, /* FC_END */
/* 864 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 866 */NdrFcShort( 0x18 ), /* 24 */
/* 868 */NdrFcShort( 0x0 ), /* 0 */
/* 870 */NdrFcShort( 0xa ), /* Offset= 10 (880) */
/* 872 */0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 874 */0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 876 */NdrFcShort( 0xffe8 ), /* Offset= -24 (852) */
/* 878 */0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 880 */
0x11, 0x0, /* FC_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 */
0x5c, /* FC_PAD */
/* 998 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 1000 */ NdrFcShort( 0x4 ), /* 4 */
/* 1002 */ NdrFcShort( 0x4 ), /* 4 */
/* 1004 */ 0x13, 0x0, /* FC_OP */
/* 1006 */ NdrFcShort( 0xffe6 ), /* Offset=-26 (980) */
/* 1008 */
0x5b, /* FC_END */

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, /* */
/* 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,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: IDispatch, ver. 0.0,

GUID={0x00020400,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* 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;
};

CINTERFACE_PROXY_VTABLE(13) _IComponentRegistrarProxyVtbl =
{
    &IComponentRegistrar_ProxyInfo,
    &IID_IComponentRegistrar,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    0 /* (void *) (INT_PTR) -1 /* IDispatch::GetTypeInfoCount */ ,
    0 /* (void *) (INT_PTR) -1 /* IDispatch::GetTypeInfo */ ,
    0 /* (void *) (INT_PTR) -1 /* IDispatch::GetIDsOfNames */ ,
    0 /* IDispatch_Invoke_Proxy */ ,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::Attach */ ,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::RegisterAll */ ,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::UnregisterAll */ ,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::GetComponents */ ,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::RegisterComponent */ ,
    (void *) (INT_PTR) -1 /* IComponentRegistrar::UnregisterComponent */
};

static const PRPC_STUB_FUNCTION IComponentRegistrar_table[] =
{
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    STUB_FORWARDING_FUNCTION,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2,
    NdrStubCall2
};

InterfaceStubVtbl _IComponentRegistrarStubVtbl =
{
    &IID_IComponentRegistrar,
    &IComponentRegistrar_ServerInfo,
    13,
    &IComponentRegistrar_table[-3],
    CStdStubBuffer_DELEGATING_METHODS
};

/* Object interface: Itpcc_com, ver. 0.0,

GUID={0x5B4FA473,0x2E68,0x4D79,{0xA6,0x26,0xF3,0x8B,0x30,0xB8,0x19,0x6E}} */

#pragma code_seg(".orpc")
static const unsigned short Itpcc_com_FormatStringOffsetTable[] =
{
    210,
    252,
    294,
    336,
    378,
    36
};

```

```

};

static const MIDL_STUBLESS_PROXY_INFO Itpcc_com_ProxyInfo =
{
    &Object_StubDesc,
    __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
};

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,
    __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
#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;
```

```
##define 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_orcs(ords_wrapper *ords,void *ctx);
extern "C" TPCCDB2GLUE_API int do_dlv(dlvy_wrapper *dlvy,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,REGISTRY_SUB_KEY
,0,KEY_READ,&registryKey) == ERROR_SUCCESS)
    {
        DEBUGMSG("Registry key open"<<endl);
        //get the null db user name
        regValueSize = sizeof(value);
        if
        (RegQueryValueEx(registryKey,DB_USER_NAME,0,&regType,(BYTE *)
&value,&regValueSize)== ERROR_SUCCESS )
            strcpy(userName,value);
        else
            return ERR_INVALID_USERNAME;
        DEBUGMSG("DB user name:"<< userName << endl);

        regValueSize = sizeof(value);
        if
        (RegQueryValueEx(registryKey,DB_USER_PASSWORD,0,&regType,(BY
TE *) &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" TPCADB2GLUE_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" TPCADB2GLUE_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" TPCADB2GLUE_API int do_pymt(payment_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" TPCADB2GLUE_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_dlvy
** 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_dlvy(dlvy_wrapper *dlvy,void *ctx)
{
    DEBUGMSG("Entered do_dlvy, 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

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
Number of frequent values retained (NUM_FREQVALUES) = 10
Number of quantiles retained (NUM_QUANTILES) = 20

Backup pending = NO

Database is consistent = YES
Rollforward pending = NO
Restore pending = NO

Multi-page file allocation enabled = NO

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) = 1000
Catalog cache size (4KB) (CATALOGCACHE_SZ) = (MAXAPPLS*4)
Log buffer size (4KB) (LOGBUFSZ) = 9216
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) = 10000

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) = 1
Number of I/O servers (NUM_IOSERVERS) = 1
Index sort flag (INDEXSORT) = YES
Sequential detect flag (SEQDETECT) = NO
Default prefetch size (pages) (DFT_PREFETCH_SZ) = 32

Track modified pages (TRACKMOD) = OFF

Default number of containers = 1
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 32

Max number of active applications (MAXAPPLS) = 1200
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/rdbfastloglv
Overflow log path (OVERFLOWLOGPATH) =
Mirror log path (MIRRORLOGPATH) =
First active log file = S0000003.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 chkpt (SOFTMAX) = 7100
Log retain for recovery enabled (LOGRETAIN) = RECOVERY
User exit for logging enabled (USEREXIT) = OFF

Auto restart enabled (AUTORESTART) = ON
Index re-creation time (INDEXREC) = SYSTEM (RESTART)
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) =

dbm.cfg.out

Database Manager Configuration

Node type = Database Server with local clients

Database manager configuration release level = 0x0a00

CPU speed (millisec/instruction) (CPUSPEED) = 3.975564e-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/java13_64

Diagnostic error capture level (DIAGLEVEL) = 1
Notify Level (NOTIFYLEVEL) = 0
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) =
 SYSMANT group name (SYSMANT_GROUP) =
 SYSMON group name (SYSMON_GROUP) =

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
 DRDA services heap size (4KB) (DRDA_HEAP_SZ) = 128

Workload impact by throttled utilities(UTIL_IMPACT_LIM) = 100

Priority of agents (AGENTPRI) = 60
 Max number of existing agents (MAXAGENTS) = 1200
 Agent pool size (NUM_POOLAGENTS) = 0
 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 (INDEXREC) = RESTART

Transaction manager database name (TM_DATABASE) = 1ST_CONN
 Transaction resync interval (sec) (RESYNC_INTERVAL) = 180

SPM name (SPM_NAME) =
 SPM log size (SPM_LOG_FILE_SZ) = 256
 SPM resync agent limit (SPM_MAX_RESYNC) = 20
 SPM log path (SPM_LOG_PATH) =

TCP/IP Service name (SVCENAME) =
 Discovery mode (DISCOVER) = SEARCH
 Discover server instance (DISCOVER_INST) = ENABLE

Maximum query degree of parallelism (MAX_QUERYDEGREE) = ANY
 Enable intra-partition parallelism (INTRA_PARALLEL) = NO

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) = 512

affinity.cfg

```
<RESOURCE_POLICY>
<METHOD>RSET</METHOD>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00000</RESOURCE>
<DBMEM_PERCENTAGE>23.4</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtpc0</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>14</NUM_CLEANERS>
<BUFFERPOOL_ID>4</BUFFERPOOL_ID>
<BUFFERPOOL_ID>8</BUFFERPOOL_ID>
<BUFFERPOOL_ID>13</BUFFERPOOL_ID>
<BUFFERPOOL_ID>17</BUFFERPOOL_ID>
<BUFFERPOOL_ID>21</BUFFERPOOL_ID>
<BUFFERPOOL_ID>25</BUFFERPOOL_ID>
<BUFFERPOOL_ID>29</BUFFERPOOL_ID>
<BUFFERPOOL_ID>33</BUFFERPOOL_ID>
<BUFFERPOOL_ID>37</BUFFERPOOL_ID>
<BUFFERPOOL_ID>41</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00001</RESOURCE>
<DBMEM_PERCENTAGE>25.96</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtpc1</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>15</NUM_CLEANERS>
<BUFFERPOOL_ID>5</BUFFERPOOL_ID>
<BUFFERPOOL_ID>9</BUFFERPOOL_ID>
<BUFFERPOOL_ID>14</BUFFERPOOL_ID>
<BUFFERPOOL_ID>18</BUFFERPOOL_ID>
<BUFFERPOOL_ID>22</BUFFERPOOL_ID>
<BUFFERPOOL_ID>26</BUFFERPOOL_ID>
<BUFFERPOOL_ID>30</BUFFERPOOL_ID>
<BUFFERPOOL_ID>34</BUFFERPOOL_ID>
<BUFFERPOOL_ID>38</BUFFERPOOL_ID>
<BUFFERPOOL_ID>42</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00002</RESOURCE>
<DBMEM_PERCENTAGE>25.96</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtpc2</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>15</NUM_CLEANERS>
<BUFFERPOOL_ID>6</BUFFERPOOL_ID>
<BUFFERPOOL_ID>10</BUFFERPOOL_ID>
<BUFFERPOOL_ID>15</BUFFERPOOL_ID>
<BUFFERPOOL_ID>19</BUFFERPOOL_ID>
<BUFFERPOOL_ID>23</BUFFERPOOL_ID>
<BUFFERPOOL_ID>27</BUFFERPOOL_ID>
<BUFFERPOOL_ID>31</BUFFERPOOL_ID>
<BUFFERPOOL_ID>35</BUFFERPOOL_ID>
<BUFFERPOOL_ID>39</BUFFERPOOL_ID>
<BUFFERPOOL_ID>43</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE>sys/node.03.00003</RESOURCE>
<DBMEM_PERCENTAGE>23.4</DBMEM_PERCENTAGE>
<SERVICE_NAME>xtpc3</SERVICE_NAME>
<BUFFERPOOL_BINDING>
<NUM_CLEANERS>14</NUM_CLEANERS>
<BUFFERPOOL_ID>7</BUFFERPOOL_ID>
```

```

<BUFFERPOOL_ID>11</BUFFERPOOL_ID>
<BUFFERPOOL_ID>16</BUFFERPOOL_ID>
<BUFFERPOOL_ID>20</BUFFERPOOL_ID>
<BUFFERPOOL_ID>24</BUFFERPOOL_ID>
<BUFFERPOOL_ID>28</BUFFERPOOL_ID>
<BUFFERPOOL_ID>32</BUFFERPOOL_ID>
<BUFFERPOOL_ID>36</BUFFERPOOL_ID>
<BUFFERPOOL_ID>40</BUFFERPOOL_ID>
<BUFFERPOOL_ID>44</BUFFERPOOL_ID>
</BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
</RESOURCE_POLICY>

```

db2set.cfg.out

```

[i] DB2_SELUDI_COMM_BUFFER=Y
[i] DB2_USE_ALTERNATE_PAGE_CLEANING=YES
[i] DB2_MAX_NON_TABLE_LOCKS=500
[i] DB2_PBT_REWRITE=ON
[i] DB2_LGPAGE_BP=yes
[i] DB2_TRUSTED_BINDIN=ON
[i] DB2_KEEPTABLELOCK=ON
[i] DB2_EVENT_LOG_CONFIG=OFF
[i] DB2_NO_FORK_CHECK=ON
[i] DB2_APM_PERFORMANCE=ALL
[i] DB2_ENABLE_BUFPD=OFF
[i] DB2_PINNED_BP=yes
[i] DB2_SELECTIVITY=YES
[i] DB2TCPCONNMGERS=1
[i] DB2ASSUMEUPDATE=ON
[i] DB2CHECKCLIENTINTERVAL=0
[i] DB2_HASH_JOIN=OFF
[i] DB2AFFINITIES=/home/tpcc/tpc-c.ibm/cfg/affinity.cfg
[i] DB2CHKSQLDA=OFF
[i] DB2ENVLIST=MEMORY_AFFINITY_DATA_SEG_SPECIAL
[i] DB2_COLLECT_TS_REC_INFO=false
[i] DB2COMM=tcPIP
[i] DB2CHKPTR=OFF

```

B.2 Transaction Monitor Parameters

tpccCom.tpcc com settings.txt

```

Transactions not supported
Enable object pooling
Minimum pool size 20
Maximum pool size 20f
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\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:000000fa
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,00,00,0,00
"MaxConnections"=dword:000061a8
"PoolThreadLimit"=dword:000000c8
"ThreadTimeout"=dword:00015180
"MaxConcurrency"=dword:ffffff

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
"Library Validation
Code"=hex:fe,5d,92,b7,40,81,c3,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,70,c1,b0,95,36,c3,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000

```

tcPIP parameters registry.reg

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TcPIP\Parameters]
"NV Hostname"="client1"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,0,0,6f,\
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"="client1"
"SearchList"="austin.ibm.com,ibm.com"
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"MaxUserPort"=dword:0000fffe
"GlobalMaxTcpWindowSize"=dword:00008000

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TcPIP\Parameters\Adapters]

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TcPIP\Parameters\Adapters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,33,00,35,00,44,00,33,00,31,00,31,00,\
42,00,46,00,2d,00,36,00,32,00,39,00,31,00,2d,00,34,00,32,00,32,00,31,00,2d,\
00,41,00,43,00,33,00,38,00,2d,00,32,00,34,00,32,00,35,00,46,00,34,00,30,00,\
45,00,41,00,45,00,37,00,38,00,7d,00,00,00,54,00,63,00,70,00,69,00,70,00,5c,\
00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,\
6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00,42,\
00,36,00,36,00,36,00,38,00,34,00,45,00,2d,00,37,00,42,00,31,00,36,00,2d,00,\

```

34,00,33,00,44,00,31,00,2d,00,39,00,33,00,36,00,46,00,2d,00,39,00,33,00,43,
\
00,37,00,37,00,33,00,31,00,33,00,39,00,36,00,38,00,37,00,7d,00,00,00,00,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:bf,11,d3,35,91,62,21,42,ac,38,24,25,f4,0e,ae,78,4e,68,66,2
b,
16,7b,d1,43,93,6f,93,c7,73,13,96,87

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{020F9853-34C4-42E8-B87B-3D04B79246C5}]
"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,00,30,00,32,00,30,00,46,00,39,00,38,00,
\
35,00,33,00,2d,00,33,00,34,00,43,00,34,00,2d,00,34,00,32,00,45,00,38,00,2d,
\
00,42,00,38,00,37,00,42,00,2d,00,33,00,44,00,30,00,34,00,42,00,37,00,39,00,
\
32,00,34,00,36,00,43,00,35,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{024BA7D5-D9EE-4502-936C-46C0B74A9971}]
"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,00,30,00,32,00,34,00,42,00,41,00,37,00,
\
44,00,35,00,2d,00,44,00,39,00,45,00,45,00,2d,00,34,00,35,00,30,00,32,00,2d,
\
00,39,00,33,00,36,00,43,00,2d,00,34,00,36,00,43,00,30,00,42,00,37,00,34,00,
\
41,00,39,00,39,00,37,00,31,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{27179934-872D-4733-B41E-696908748FB3}]
"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,00,32,00,37,00,31,00,37,00,39,00,39,00,
\
33,00,34,00,2d,00,38,00,37,00,32,00,44,00,2d,00,34,00,37,00,33,00,33,00,2d,
\
00,42,00,34,00,31,00,45,00,2d,00,36,00,39,00,36,00,39,00,30,00,38,00,37,00,
\
34,00,38,00,46,00,42,00,33,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{36A91812-DCB7-401D-B651-D1BE9F94EE35}]
"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,00,33,00,36,00,41,00,39,00,31,00,38,00,
\
31,00,32,00,2d,00,44,00,43,00,42,00,37,00,2d,00,34,00,30,00,31,00,44,00,2d,
\
00,42,00,36,00,35,00,31,00,2d,00,44,00,31,00,42,00,45,00,39,00,46,00,39,00,46,00,39,00,
\
34,00,45,00,45,00,33,00,35,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{4D3754BC-6467-4048-A9C5-5DDD789623BC}]
"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,00,34,00,44,00,33,00,37,00,35,00,34,00,
\
42,00,43,00,2d,00,36,00,34,00,36,00,37,00,2d,00,34,00,30,00,34,00,38,00,2d,
\
00,41,00,39,00,43,00,35,00,2d,00,35,00,44,00,44,00,44,00,37,00,38,00,39,00,
\
36,00,32,00,33,00,42,00,43,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{58927A28-871D-4702-B1BB-F90A85850522}]
"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,00,35,00,38,00,39,00,32,00,37,00,41,00,
\
32,00,38,00,2d,00,38,00,37,00,31,00,44,00,2d,00,34,00,37,00,30,00,32,00,2d,
\
00,42,00,31,00,42,00,42,00,2d,00,46,00,39,00,30,00,41,00,38,00,35,00,38,00,
\
35,00,30,00,35,00,32,00,32,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{A0005D8B-DBAF-47F8-B9A0-D438E7D5FE06}]
"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,00,41,00,30,00,30,00,30,00,35,00,44,00,
\
38,00,42,00,2d,00,44,00,42,00,41,00,46,00,2d,00,34,00,37,00,46,00,38,00,2d,
\
00,42,00,39,00,41,00,30,00,2d,00,44,00,34,00,33,00,38,00,45,00,37,00,44,00,
\
35,00,46,00,45,00,30,00,36,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{B7C9E08E-C9FA-4D6A-B2FE-1EE47860ACFB}]
"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,00,42,00,37,00,43,00,39,00,45,00,30,00,
\
38,00,45,00,2d,00,43,00,39,00,46,00,41,00,2d,00,34,00,44,00,36,00,41,00,2d,
\
00,42,00,32,00,46,00,45,00,2d,00,31,00,45,00,45,00,34,00,37,00,38,00,36,00,
\
30,00,41,00,43,00,46,00,42,00,7d,00,00,00,00,00

00,42,00,36,00,35,00,31,00,2d,00,44,00,31,00,42,00,45,00,39,00,46,00,39,00,
\
34,00,45,00,45,00,33,00,35,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{4D3754BC-6467-4048-A9C5-5DDD789623BC}]
"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,00,34,00,44,00,33,00,37,00,35,00,34,00,
\
42,00,43,00,2d,00,36,00,34,00,36,00,37,00,2d,00,34,00,30,00,34,00,38,00,2d,
\
00,41,00,39,00,43,00,35,00,2d,00,35,00,44,00,44,00,44,00,37,00,38,00,39,00,
\
36,00,32,00,33,00,42,00,43,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{58927A28-871D-4702-B1BB-F90A85850522}]
"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,00,35,00,38,00,39,00,32,00,37,00,41,00,
\
32,00,38,00,2d,00,38,00,37,00,31,00,44,00,2d,00,34,00,37,00,30,00,32,00,2d,
\
00,42,00,31,00,42,00,42,00,2d,00,46,00,39,00,30,00,41,00,38,00,35,00,38,00,
\
35,00,30,00,35,00,32,00,32,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{A0005D8B-DBAF-47F8-B9A0-D438E7D5FE06}]
"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,00,41,00,30,00,30,00,30,00,35,00,44,00,
\
38,00,42,00,2d,00,44,00,42,00,41,00,46,00,2d,00,34,00,37,00,46,00,38,00,2d,
\
00,42,00,39,00,41,00,30,00,2d,00,44,00,34,00,33,00,38,00,45,00,37,00,44,00,
\
35,00,46,00,45,00,30,00,36,00,7d,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\P
arameters\Adapters\{B7C9E08E-C9FA-4D6A-B2FE-1EE47860ACFB}]
"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,00,42,00,37,00,43,00,39,00,45,00,30,00,
\
38,00,45,00,2d,00,43,00,39,00,46,00,41,00,2d,00,34,00,44,00,36,00,41,00,2d,
\
00,42,00,32,00,46,00,45,00,2d,00,31,00,45,00,45,00,34,00,37,00,38,00,36,00,
\
30,00,41,00,43,00,46,00,42,00,7d,00,00,00,00,00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{BF58EED1-8711-4672-9E84-4D9983C0FCDE}]
"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,00,42,00,46,00,35,00,38,00,45,00,45,00,\
44,00,31,00,2d,00,38,00,37,00,31,00,31,00,2d,00,34,00,36,00,37,00,32,00,2d,\
\
00,39,00,45,00,38,00,34,00,2d,00,34,00,44,00,39,00,39,00,38,00,33,00,43,00,\
30,00,46,00,43,00,44,00,45,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{C20540AD-2C0D-4EEA-9F16-13B14AC3A745}]
"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,00,43,00,32,00,30,00,35,00,34,00,30,00,\
41,00,44,00,2d,00,32,00,43,00,30,00,44,00,2d,00,34,00,45,00,45,00,41,00,2d,\
\
00,39,00,46,00,31,00,36,00,2d,00,31,00,33,00,42,00,31,00,34,00,41,00,43,00,\
33,00,41,00,37,00,34,00,35,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{CB128FF9-BDD9-4072-A2EA-CB2C610C47F3}]
"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,00,43,00,42,00,31,00,32,00,38,00,46,00,\
46,00,39,00,2d,00,42,00,44,00,44,00,39,00,2d,00,34,00,30,00,37,00,32,00,2d,\
\
00,41,00,32,00,45,00,41,00,2d,00,43,00,42,00,32,00,43,00,36,00,31,00,30,00,\
43,00,34,00,37,00,46,00,33,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{D7359E7D-56BC-4040-8071-4EB6C9C56819}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,44,00,37,00,33,00,35,00,39,00,45,00,\
37,00,44,00,2d,00,35,00,36,00,42,00,43,00,2d,00,34,00,30,00,34,00,30,00,2d,\
\
00,38,00,30,00,37,00,31,00,2d,00,34,00,45,00,42,00,36,00,43,00,39,00,43,00,\
35,00,36,00,38,00,31,00,39,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{F0FB9B9D-935B-4532-A718-430EBB4D9D57}]
```

```
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,46,00,30,00,46,00,42,00,42,00,42,00,\
39,00,44,00,2d,00,39,00,33,00,35,00,42,00,2d,00,34,00,35,00,33,00,32,00,2d,\
\
00,41,00,37,00,31,00,38,00,2d,00,34,00,33,00,30,00,45,00,42,00,42,00,34,00,\
44,00,39,00,44,00,35,00,37,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{F3FBC49C-658F-4C39-87D1-3B05BBFB78D5}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,46,00,33,00,46,00,42,00,43,00,34,00,\
39,00,43,00,2d,00,36,00,35,00,38,00,46,00,2d,00,34,00,43,00,33,00,39,00,2d,\
\
00,38,00,37,00,44,00,31,00,2d,00,33,00,42,00,30,00,35,00,42,00,42,00,46,00,\
42,00,37,00,38,00,44,00,35,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{FD034174-E406-408F-9FDE-093A285D30FD}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\
00,61,00,63,00,65,00,73,00,5c,00,7b,00,46,00,44,00,30,00,33,00,34,00,31,00,\
37,00,34,00,2d,00,45,00,34,00,30,00,36,00,2d,00,34,00,30,00,38,00,46,00,2d,\
\
00,39,00,46,00,44,00,45,00,2d,00,30,00,39,00,33,00,41,00,32,00,38,00,35,00,\
44,00,33,00,30,00,46,00,44,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DNSRegisteredAdapters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{020F9853-34C4-42E8-B87B-3D04B79246C5}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"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
```


"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,
"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):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,
0,00,\
35,00,00,00,00,00,
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3fef8b28
"T1"=dword:3fef9230
"T2"=dword:3fef9776
"LeaseTerminatesTime"=dword:3fef9938
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{58927A28-871D-4702-B1BB-F90A85850522}]\
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):39,00,2e,00,34,00,31,00,2e,00,39,00,30,00,2e,00,31,00,34,00,
4,00,\
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,
"DefaultGateway"=hex(7):39,00,2e,00,34,00,31,00,2e,00,39,00,30,00,2e,00,31,00,
1,00,\
00,00,00,00,
"DefaultGatewayMetric"=hex(7):31,00,00,00,00,00,
"NameServer"="9.0.7.1"
"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):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,
0,00,\
34,00,00,00,00,00,
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3f6f437c
"T1"=dword:3f6f4a84
"T2"=dword:3f6f4fca
"LeaseTerminatesTime"=dword:3f6f518c
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"DhcpClassIdBin"=hex:

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A0005D8B-DBAF-47F8-B9A0-D438E7D5FE06}]\
"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,00,32,00,00,00,
0,00,
"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,
"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):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,
0,00,\
34,00,00,00,00,00,
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3fef86ca
"T1"=dword:3fef8dd2
"T2"=dword:3fef9318
"LeaseTerminatesTime"=dword:3fef94da
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"TcpWindowSize"=dword:00008000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{B7C9E08E-C9FA-4D6A-B2FE-1EE47860ACFB}]\
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,35,00,2e,00,31,00,00,00,
0,00,
"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,
"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):30,00,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,
0,00,\
33,00,00,00,00,00,
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3f6f4d69
"T1"=dword:3f6f5471
"T2"=dword:3f6f59b7
"LeaseTerminatesTime"=dword:3f6f5b79
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"DhcpClassIdBin"=hex:

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{BF58EED1-8711-4672-9E84-4D9983C0FCDE}]\
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000

```
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,33,00,2e,00,31,00,00,00,0,00
"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):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
39,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3f6f4b43
"T1"=dword:3f6f524b
"T2"=dword:3f6f5791
"LeaseTerminatesTime"=dword:3f6f5953
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"DhcpClassIdBin"=hex:
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{C20540AD-2C0D-4EEA-9F16-13B14AC3A745}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,00,31,00,30,00,30,00,2e,00,34,00,35,00,\
2e,00,32,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"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
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3fe76beb
"T1"=dword:3fe772f3
"T2"=dword:3fe77839
"LeaseTerminatesTime"=dword:3fe779fb
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{CB128FF9-BDD9-4072-A2EA-CB2C610C47F3}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
```

```
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,30,00,2e,00,32,00,00,00,0,00
"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):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
37,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3f6f4c6c
"T1"=dword:3f6f5374
"T2"=dword:3f6f58ba
"LeaseTerminatesTime"=dword:3f6f5a7c
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"DhcpClassIdBin"=hex:
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{D7359E7D-56BC-4040-8071-4EB6C9C56819}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,36,00,2e,00,31,00,00,00,0,00
"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):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\
35,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3f6f4d44
"T1"=dword:3f6f54d4
"T2"=dword:3f6f5a1a
"LeaseTerminatesTime"=dword:3f6f5bdc
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
"DhcpClassIdBin"=hex:
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{F0FBBB9D-935B-4532-A718-430EBB4D9D57}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):39,00,2e,00,33,00,2e,00,38,00,33,00,2e,00,31,00,32,00,32,00,\
```

```
00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,32,00,32,00,34,00,00,00,00,00
"DefaultGateway"=hex(7):39,00,2e,00,33,00,2e,00,38,00,33,00,2e,00,39,00,37,00,\
00,00,00,00
"DefaultGatewayMetric"=hex(7):31,00,00,00,00,00,00
"NameServer"="9.0.7.1,9.0.6.11"
"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,\
00,00,\
36,00,00,00,00,00,00
"DhcpClassIdBin"=hex:\
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3fc27478
"T1"=dword:3fc27b80
"T2"=dword:3fc280c6
"LeaseTerminatesTime"=dword:3fc28288
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{F3FBC49C-658F-4C39-87D1-3B05BBFB78D5}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00,00
"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,\
00,00,\
32,00,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{FD034174-E406-408F-9FDE-093A285D30FD}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,30,00,2e,00,31,00,2e,00,31,00,2e,00,31,00,00,00,\
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,00,30,00,00,00,00,00,00
"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,\
00,00,\
32,00,00,00,00,00,00
"DhcpClassIdBin"=hex:\
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:3fe7ab37
"T1"=dword:3fe7b23f
"T2"=dword:3fe7b785
"LeaseTerminatesTime"=dword:3fe7b947
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:a73c1fca
"AddressType"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,\
6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,\
00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,00,6c,00,6c,00,\
00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,00,\
02,\
00,00,00,01,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00,00,\
00,00,00,00,00,06,00,00,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,00,02,00,00,\
00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00
```

Tpcc software registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\TPCC]
"dIvyLogPath"="c:\\inetpub\\wwwroot\\tpcc\\dIvy"
"dIvyQueueLen"=dword:00004e20
"dIvyThreads"=dword:00000005
>nullDB"=dword:00000000
"htmlTrace"=dword:00000000
"dbName"="tpcc"

"errorLogFile"="c:\\inetpub\\wwwroot\\tpcc\\errorLog.txt"
"htmlTraceLogFile"="c:\\inetpub\\wwwroot\\tpcc\\htmlTrace.txt"
"numUsers"=dword:000061a8
"dbType"="DB2"
"dbUserName"="tpcc"
"dbPassword"="tpcc"
"dbInterfacePath"="C:\\inetpub\\wwwroot\\tpcc\\db2glue.dll"
```

W3SVC registry.reg

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):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,00,65,00,74,00,73,\
00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,00,6f,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,00,49,00,4e,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,00,00,00,00,00,00,00,00,03,00,00,00,78,ad,14,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP]
```

```
"NOTE"="This is for backward compatibility only."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
```

```
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"CertMapList"="C:\\WINNT\\System32\\inetsrv\\iisrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]
```

```
"/"="c:\\inetpub\\wwwroot,,201"
"/Scripts"="c:\\inetpub\\scripts,,204"
"/IISHelp"="c:\\winnt\\help\\iishelp,,201"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\\iisadmin,,201"
"/IISamples"="c:\\inetpub\\iissamples,,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"
"/PBServer"="C:\\Program Files\\Phone Book Service\\Bin,,5"
"/PBServerData"="C:\\Program Files\\Phone Book Service\\Data,,1"
"/Rpc"="C:\\WINNT\\System32\\RpcProxy,,4"
"/Printers"="C:\\WINNT\\web\\printers,,201"
"/tpcc"="C:\\inetpub\\wwwroot\\tpcc,,207"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\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:fa,8b,c6,5d,41,81,c3,01,10,1d,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,70,c1,b0,95,36,c3,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
```

```
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
```

```
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,05,\
```

```
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
\
```

```
00,00,05,0b,00,00,00,20,02,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
```

```
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,05,12,00,00,\
```

```
\
00,01,01,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
```

```
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

B.3 AIX Parameters

IBM eServer pSeries 690

SW_dist_intr	false	Enable SW distribution of interrupts	True
autorestart	true	Automatically REBOOT system after a crash	True
boottype	disk	N/A	False
conslogin	enable	System Console Login	False
cpuguard	enable	CPU Guard	True
frequency	634000000	System Bus Frequency	False
fullcore	false	Enable full CORE dump	True
fwversion	IBM,RG031002_regatta	Firmware version and revision levels	False
iostat	false	Continuously maintain DISK I/O history	True
keylock	normal	State of system keylock at boot time	False
maxbuf	20	Maximum number of pages in block I/O BUFFER	
CACHE	True		
maxmbuf	0	Maximum Kbytes of real memory allowed for MBUFS	True
maxpout	0	HIGH water mark for pending write I/Os per file	True
maxuproc	60000	Maximum number of PROCESSES allowed per user	True
minpout	0	LOW water mark for pending write I/Os per file	True
modelName	IBM,7040-681	Machine name	False
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	1073741824	Amount of usable physical memory in Kbytes	False

```

rtasversion 1      Open Firmware RTAS version      False
systemid  IBM,010ykrh0v  Hardware system identifier      False

```

vmo -I

NAME TYPE	CUR	DEF	BOOT	MIN	MAX	UNIT	
DEPENDENCIES							
memory_frames	256M	256M		4KB	pages		S
pinnable_frames	8353K	8353K		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%	1	20	1	1	100	% memory	D
maxperm%							
minperm	85627	85627					S
maxperm%	80	80	80	1	100	% memory	D
minperm%							
maxclient%							
maxperm	6689K	6689K					S
strict_maxperm	0	0	0	0	1	boolean	D
maxpin%	98	99	98	1	99	% memory	D
pinnable_frames							
memory_frames							
maxpin	256901K	256901K					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	4095M	uid	D
numpsblks	31696K	31696K				4KB pages	S
npskill	253568	253568	253568	1	31695K	4KB pages	D
npswarn	990K	990K	990K	0	31695K	4KB pages	D
v_pinshm	1	0	1	0	1	boolean	D
pta_balance_threshold	n/a	50	50	1	99	% pta segment	R
pagecoloring	n/a	0	0	0	1	boolean	B
framesets	2	2	2	1	10		B
mempools	1	1	1	1	32		B
lgpg_size	16M	0	16M	0	256M	bytes	B
lgpg_regions							
lgpg_regions	61500	0	61500	0			B
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	-4	-1	-4	-4	0		B
force_realias_lite	0	0	0	0	1	boolean	D
realias_percentage	0	0	0	0	32767		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

ioo -L

NAME TYPE	CUR	DEF	BOOT	MIN	MAX	UNIT	
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	2047M	16KB/cluster	D
numfsbufs	196	196	196	1	2047M		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	128	4KB pages	D
j2_maxPageReadAhead	8	8	8	0	128	4KB pages	D
j2_nBufferPerPagerDevice	512	512	512	0	2047M		M
j2_nPagesPerWriteBehindCluster	32	32	32	0	128		D
j2_maxRandomWrite	0	0	0	0	128	4KB pages	D
j2_nRandomCluster	0	0	0	0	2047M	16KB clusters	D
hd_pvs_opn	100	100					S
hd_pbuf_cnt	15616	13K	13K	0	2047M		I

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: 2^{10}
- G = Giga: 2^{30}
- P = Peta: 2^{50}
- M = Mega: 2^{20}
- T = Tera: 2^{40}
- E = Exa: 2^{60}

Appendix C: Database Setup Code

C1. Database Creation Scripts

db/create_database.ddl

```
create database tpcc on /home/tpcc/db/tpccdb1 collate using identity
catalog tablespace managed by system using ('/home/tpcc/db/db1catalog');
```

ts/crts_cust.ddl

```
connect to tpcc;
```

```
-- now creating TS for ts_customer_01 of D1
```

```
drop tablespace ts_customer_01;
create regular tablespace ts_customer_01 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1CST01' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_02 of D1
```

```
drop tablespace ts_customer_02;
create regular tablespace ts_customer_02 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V2CST02' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_03 of D1
```

```
drop tablespace ts_customer_03;
create regular tablespace ts_customer_03 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V3CST03' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_04 of D1
```

```
drop tablespace ts_customer_04;
create regular tablespace ts_customer_04 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V1CST04' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_05 of D1
```

```
drop tablespace ts_customer_05;
create regular tablespace ts_customer_05 pagesize 4K
```

```
  managed by database
  using
  (
    device '/dev/rD1F02V2CST05' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_06 of D1
```

```
drop tablespace ts_customer_06;
create regular tablespace ts_customer_06 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V3CST06' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_07 of D1
```

```
drop tablespace ts_customer_07;
create regular tablespace ts_customer_07 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V1CST07' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_08 of D1
```

```
drop tablespace ts_customer_08;
create regular tablespace ts_customer_08 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V2CST08' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_09 of D1
```

```
drop tablespace ts_customer_09;
create regular tablespace ts_customer_09 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3CST09' 13558016
  )
  extentsize 256
  prefetchsize 512;
commit;
```

```
-- now creating TS for ts_customer_10 of D1
```

```
drop tablespace ts_customer_10;
create regular tablespace ts_customer_10 pagesize 4K
  managed by database
  using
  (
```



```

        device '/dev/rD1F04V1CST10' 13558016
    )
    extentsize 256
    prefetchsize 512;
commit;

-- now creating TS for ts_customer_11 of D1

drop tablespace ts_customer_11;
create regular tablespace ts_customer_11 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V2CST11' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_12 of D1

drop tablespace ts_customer_12;
create regular tablespace ts_customer_12 pagesize 4K
managed by database
using
(
    device '/dev/rD1F04V3CST12' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_13 of D1

drop tablespace ts_customer_13;
create regular tablespace ts_customer_13 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1CST13' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_14 of D1

drop tablespace ts_customer_14;
create regular tablespace ts_customer_14 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2CST14' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_15 of D1

drop tablespace ts_customer_15;
create regular tablespace ts_customer_15 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3CST15' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_16 of D1

```

```

drop tablespace ts_customer_16;
create regular tablespace ts_customer_16 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1CST16' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_17 of D1

drop tablespace ts_customer_17;
create regular tablespace ts_customer_17 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2CST17' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_18 of D1

drop tablespace ts_customer_18;
create regular tablespace ts_customer_18 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3CST18' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_19 of D1

drop tablespace ts_customer_19;
create regular tablespace ts_customer_19 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1CST19' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_20 of D1

drop tablespace ts_customer_20;
create regular tablespace ts_customer_20 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2CST20' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_21 of D1

drop tablespace ts_customer_21;
create regular tablespace ts_customer_21 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3CST21' 13558016
)

```

```

)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_22 of D1

drop tablespace ts_customer_22;
create regular tablespace ts_customer_22 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V1CST22' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_23 of D1

drop tablespace ts_customer_23;
create regular tablespace ts_customer_23 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V2CST23' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_24 of D1

drop tablespace ts_customer_24;
create regular tablespace ts_customer_24 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V3CST24' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_25 of D1

drop tablespace ts_customer_25;
create regular tablespace ts_customer_25 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V1CST25' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_26 of D1

drop tablespace ts_customer_26;
create regular tablespace ts_customer_26 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V2CST26' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_27 of D1

```

```

drop tablespace ts_customer_27;
create regular tablespace ts_customer_27 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V3CST27' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_28 of D1

drop tablespace ts_customer_28;
create regular tablespace ts_customer_28 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V1CST28' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_29 of D1

drop tablespace ts_customer_29;
create regular tablespace ts_customer_29 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V2CST29' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_30 of D1

drop tablespace ts_customer_30;
create regular tablespace ts_customer_30 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V3CST30' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_31 of D1

drop tablespace ts_customer_31;
create regular tablespace ts_customer_31 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V1CST31' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_32 of D1

drop tablespace ts_customer_32;
create regular tablespace ts_customer_32 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V2CST32' 13558016
)

```

```

    extentsize 256
    prefetchsize 512;
commit;

-- now creating TS for ts_customer_33 of D1

drop tablespace ts_customer_33;
create regular tablespace ts_customer_33 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V3CST33' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_34 of D1

drop tablespace ts_customer_34;
create regular tablespace ts_customer_34 pagesize 4K
managed by database
using
(
    device '/dev/rD1F12V1CST34' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_35 of D1

drop tablespace ts_customer_35;
create regular tablespace ts_customer_35 pagesize 4K
managed by database
using
(
    device '/dev/rD1F12V2CST35' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_36 of D1

drop tablespace ts_customer_36;
create regular tablespace ts_customer_36 pagesize 4K
managed by database
using
(
    device '/dev/rD1F12V3CST36' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_37 of D1

drop tablespace ts_customer_37;
create regular tablespace ts_customer_37 pagesize 4K
managed by database
using
(
    device '/dev/rD1F13V1CST37' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for ts_customer_38 of D1

drop tablespace ts_customer_38;

```

```

create regular tablespace ts_customer_38 pagesize 4K
managed by database
using
(
    device '/dev/rD1F13V2CST38' 13558016
)
extentsize 256
prefetchsize 512;
commit;

-- now creating TS for is_customer_01 of D1

drop tablespace is_customer_01;
create regular tablespace is_customer_01 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V1CSTI01' 430080
)
extentsize 256
prefetchsize 512
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_02 of D1

drop tablespace is_customer_02;
create regular tablespace is_customer_02 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V2CSTI02' 430080
)
extentsize 256
prefetchsize 512
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_03 of D1

drop tablespace is_customer_03;
create regular tablespace is_customer_03 pagesize 8K
managed by database
using
(
    device '/dev/rD1F01V3CSTI03' 430080
)
extentsize 256
prefetchsize 512
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_04 of D1

drop tablespace is_customer_04;
create regular tablespace is_customer_04 pagesize 8K
managed by database
using
(
    device '/dev/rD1F02V1CSTI04' 430080
)
extentsize 256
prefetchsize 512
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_05 of D1

drop tablespace is_customer_05;
create regular tablespace is_customer_05 pagesize 8K
managed by database
using

```

```

(
  device '/dev/rD1F02V2CSTI05' 430080
)
extentsize 256
prefetchsize 512
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_06 of D1

drop tablespace is_customer_06;
create regular tablespace is_customer_06 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V3CSTI06' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_07 of D1

drop tablespace is_customer_07;
create regular tablespace is_customer_07 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V1CSTI07' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_08 of D1

drop tablespace is_customer_08;
create regular tablespace is_customer_08 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V2CSTI08' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_09 of D1

drop tablespace is_customer_09;
create regular tablespace is_customer_09 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V3CSTI09' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_10 of D1

drop tablespace is_customer_10;
create regular tablespace is_customer_10 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V1CSTI10' 430080

```

```

)
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_11 of D1

drop tablespace is_customer_11;
create regular tablespace is_customer_11 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V2CSTI11' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_12 of D1

drop tablespace is_customer_12;
create regular tablespace is_customer_12 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V3CSTI12' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_13 of D1

drop tablespace is_customer_13;
create regular tablespace is_customer_13 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V1CSTI13' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_14 of D1

drop tablespace is_customer_14;
create regular tablespace is_customer_14 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V2CSTI14' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_15 of D1

drop tablespace is_customer_15;
create regular tablespace is_customer_15 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V3CSTI15' 430080
  )
  extentsize 256

```

```

    prefetchsize 512
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_16 of D1

drop tablespace is_customer_16;
create regular tablespace is_customer_16 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V1CSTI16' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_17 of D1

drop tablespace is_customer_17;
create regular tablespace is_customer_17 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V2CSTI17' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_18 of D1

drop tablespace is_customer_18;
create regular tablespace is_customer_18 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V3CSTI18' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_19 of D1

drop tablespace is_customer_19;
create regular tablespace is_customer_19 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V1CSTI19' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_20 of D1

drop tablespace is_customer_20;
create regular tablespace is_customer_20 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V2CSTI20' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;

```

```

commit;

-- now creating TS for is_customer_21 of D1

drop tablespace is_customer_21;
create regular tablespace is_customer_21 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V3CSTI21' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_22 of D1

drop tablespace is_customer_22;
create regular tablespace is_customer_22 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V1CSTI22' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_23 of D1

drop tablespace is_customer_23;
create regular tablespace is_customer_23 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V2CSTI23' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_24 of D1

drop tablespace is_customer_24;
create regular tablespace is_customer_24 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V3CSTI24' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_25 of D1

drop tablespace is_customer_25;
create regular tablespace is_customer_25 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V1CSTI25' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_customer_26 of D1

drop tablespace is_customer_26;
create regular tablespace is_customer_26 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V2CSTI26' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_27 of D1

drop tablespace is_customer_27;
create regular tablespace is_customer_27 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V3CSTI27' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_28 of D1

drop tablespace is_customer_28;
create regular tablespace is_customer_28 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V1CSTI28' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_29 of D1

drop tablespace is_customer_29;
create regular tablespace is_customer_29 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V2CSTI29' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_30 of D1

drop tablespace is_customer_30;
create regular tablespace is_customer_30 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V3CSTI30' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_31 of D1

```

```

drop tablespace is_customer_31;
create regular tablespace is_customer_31 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V1CSTI31' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_32 of D1

drop tablespace is_customer_32;
create regular tablespace is_customer_32 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V2CSTI32' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_33 of D1

drop tablespace is_customer_33;
create regular tablespace is_customer_33 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V3CSTI33' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_34 of D1

drop tablespace is_customer_34;
create regular tablespace is_customer_34 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V1CSTI34' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_35 of D1

drop tablespace is_customer_35;
create regular tablespace is_customer_35 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V2CSTI35' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_36 of D1

```

```

drop tablespace is_customer_36;
create regular tablespace is_customer_36 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V3CSTI36' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for is_customer_37 of D1

```

drop tablespace is_customer_37;
create regular tablespace is_customer_37 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F13V1CSTI37' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for is_customer_38 of D1

```

drop tablespace is_customer_38;
create regular tablespace is_customer_38 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F13V2CSTI38' 430080
  )
  extentsize 256
  prefetchsize 512
  bufferpool ibmdefaultbp8K;
commit;

```

connect reset;

ts/crts_dist.ddl

connect to tpcc;

-- now creating TS for ts_dist_01 of D1

```

drop tablespace ts_dist_01;
create regular tablespace ts_dist_01 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

-- now creating TS for ts_dist_02 of D1

```

drop tablespace ts_dist_02;
create regular tablespace ts_dist_02 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

-- now creating TS for ts_dist_03 of D1

```

drop tablespace ts_dist_03;
create regular tablespace ts_dist_03 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

-- now creating TS for ts_dist_04 of D1

```

drop tablespace ts_dist_04;
create regular tablespace ts_dist_04 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

-- now creating TS for ts_dist_05 of D1

```

drop tablespace ts_dist_05;
create regular tablespace ts_dist_05 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

-- now creating TS for ts_dist_06 of D1

```

drop tablespace ts_dist_06;
create regular tablespace ts_dist_06 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

-- now creating TS for ts_dist_07 of D1

```

drop tablespace ts_dist_07;
create regular tablespace ts_dist_07 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

-- now creating TS for ts_dist_08 of D1

```

drop tablespace ts_dist_08;
create regular tablespace ts_dist_08 pagesize 4K
  managed by database
  using

```

```

(
  device '/dev/rD1F03V2DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_09 of D1

drop tablespace ts_dist_09;
create regular tablespace ts_dist_09 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_10 of D1

drop tablespace ts_dist_10;
create regular tablespace ts_dist_10 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_11 of D1

drop tablespace ts_dist_11;

create regular tablespace ts_dist_11 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_12 of D1

drop tablespace ts_dist_12;
create regular tablespace ts_dist_12 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_13 of D1

drop tablespace ts_dist_13;
create regular tablespace ts_dist_13 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

```

-- now creating TS for ts_dist_14 of D1

drop tablespace ts_dist_14;
create regular tablespace ts_dist_14 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_15 of D1

drop tablespace ts_dist_15;
create regular tablespace ts_dist_15 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_16 of D1

drop tablespace ts_dist_16;
create regular tablespace ts_dist_16 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_17 of D1

drop tablespace ts_dist_17;
create regular tablespace ts_dist_17 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_18 of D1

drop tablespace ts_dist_18;
create regular tablespace ts_dist_18 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_19 of D1

drop tablespace ts_dist_19;
create regular tablespace ts_dist_19 pagesize 4K
  managed by database
  using

```



```

(
  device '/dev/rD1F07V1DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_20 of D1

drop tablespace ts_dist_20;
create regular tablespace ts_dist_20 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_21 of D1

drop tablespace ts_dist_21;
create regular tablespace ts_dist_21 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_22 of D1

drop tablespace ts_dist_22;
create regular tablespace ts_dist_22 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_23 of D1

drop tablespace ts_dist_23;
create regular tablespace ts_dist_23 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_24 of D1

drop tablespace ts_dist_24;
create regular tablespace ts_dist_24 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

```

```

-- now creating TS for ts_dist_25 of D1

drop tablespace ts_dist_25;
create regular tablespace ts_dist_25 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_26 of D1

drop tablespace ts_dist_26;
create regular tablespace ts_dist_26 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_27 of D1

drop tablespace ts_dist_27;
create regular tablespace ts_dist_27 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V3DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_28 of D1

drop tablespace ts_dist_28;
create regular tablespace ts_dist_28 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V1DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_29 of D1

drop tablespace ts_dist_29;
create regular tablespace ts_dist_29 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V2DIST' 1536
  )
  extentsize 64
  prefetchsize 2048;
commit;

-- now creating TS for ts_dist_30 of D1

drop tablespace ts_dist_30;
create regular tablespace ts_dist_30 pagesize 4K
  managed by database
  using
  (

```

```

        device '/dev/rD1F10V3DIST' 1536
    )
    extentsize 64
    prefetchsize 2048;
commit;

-- now creating TS for ts_dist_31 of D1

drop tablespace ts_dist_31;
create regular tablespace ts_dist_31 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V1DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_32 of D1

drop tablespace ts_dist_32;
create regular tablespace ts_dist_32 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V2DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_33 of D1

drop tablespace ts_dist_33;
create regular tablespace ts_dist_33 pagesize 4K
managed by database
using
(
    device '/dev/rD1F11V3DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_34 of D1

drop tablespace ts_dist_34;
create regular tablespace ts_dist_34 pagesize 4K
managed by database
using
(
    device '/dev/rD1F12V1DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_35 of D1

drop tablespace ts_dist_35;
create regular tablespace ts_dist_35 pagesize 4K
managed by database
using
(
    device '/dev/rD1F12V2DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_36 of D1

```

```

drop tablespace ts_dist_36;
create regular tablespace ts_dist_36 pagesize 4K
managed by database
using
(
    device '/dev/rD1F12V3DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_37 of D1

drop tablespace ts_dist_37;
create regular tablespace ts_dist_37 pagesize 4K
managed by database
using
(
    device '/dev/rD1F13V1DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

-- now creating TS for ts_dist_38 of D1

drop tablespace ts_dist_38;
create regular tablespace ts_dist_38 pagesize 4K
managed by database
using
(
    device '/dev/rD1F13V2DIST' 1536
)
extentsize 64
prefetchsize 2048;
commit;

connect reset;

```

ts/crts_hist.ddl

```

connect to tpc;

-- now creating TS for ts_history_01 of D1

drop tablespace ts_history_01;
create regular tablespace ts_history_01 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_02 of D1

drop tablespace ts_history_02;
create regular tablespace ts_history_02 pagesize 16K
managed by database
using
(
    device '/dev/rD1F01V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;

```

```

commit;

-- now creating TS for ts_history_03 of D1

drop tablespace ts_history_03;
create regular tablespace ts_history_03 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F01V3HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_04 of D1

drop tablespace ts_history_04;
create regular tablespace ts_history_04 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F02V1HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_05 of D1

drop tablespace ts_history_05;
create regular tablespace ts_history_05 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F02V2HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_06 of D1

drop tablespace ts_history_06;
create regular tablespace ts_history_06 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F02V3HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_07 of D1

drop tablespace ts_history_07;
create regular tablespace ts_history_07 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F03V1HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_08 of D1

drop tablespace ts_history_08;
create regular tablespace ts_history_08 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F03V2HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_09 of D1

drop tablespace ts_history_09;
create regular tablespace ts_history_09 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F03V3HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_10 of D1

drop tablespace ts_history_10;
create regular tablespace ts_history_10 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F04V1HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_11 of D1

drop tablespace ts_history_11;
create regular tablespace ts_history_11 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F04V2HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_12 of D1

drop tablespace ts_history_12;
create regular tablespace ts_history_12 pagesize 16K
  managed by database
  using
  (
    device '/dev/rD1F04V3HIST' 394496
  )
  extentsize 256
  prefetchsize 2048
  bufferpool ibmdefaultbp16K;
commit;

```

```

-- now creating TS for ts_history_13 of D1

drop tablespace ts_history_13;
create regular tablespace ts_history_13 pagesize 16K
managed by database
using
(
  device '/dev/rD1F05V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_14 of D1

drop tablespace ts_history_14;
create regular tablespace ts_history_14 pagesize 16K
managed by database
using
(
  device '/dev/rD1F05V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_15 of D1

drop tablespace ts_history_15;
create regular tablespace ts_history_15 pagesize 16K
managed by database
using
(
  device '/dev/rD1F05V3HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_16 of D1

drop tablespace ts_history_16;
create regular tablespace ts_history_16 pagesize 16K
managed by database
using
(
  device '/dev/rD1F06V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_17 of D1

drop tablespace ts_history_17;
create regular tablespace ts_history_17 pagesize 16K
managed by database
using
(
  device '/dev/rD1F06V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_18 of D1

```

```

drop tablespace ts_history_18;
create regular tablespace ts_history_18 pagesize 16K
managed by database
using
(
  device '/dev/rD1F06V3HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_19 of D1

drop tablespace ts_history_19;
create regular tablespace ts_history_19 pagesize 16K
managed by database
using
(
  device '/dev/rD1F07V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_20 of D1

drop tablespace ts_history_20;
create regular tablespace ts_history_20 pagesize 16K
managed by database
using
(
  device '/dev/rD1F07V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_21 of D1

drop tablespace ts_history_21;
create regular tablespace ts_history_21 pagesize 16K
managed by database
using
(
  device '/dev/rD1F07V3HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_22 of D1

drop tablespace ts_history_22;
create regular tablespace ts_history_22 pagesize 16K
managed by database
using
(
  device '/dev/rD1F08V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_23 of D1

drop tablespace ts_history_23;
create regular tablespace ts_history_23 pagesize 16K

```

```

managed by database
using
(
  device '/dev/rD1F08V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_24 of D1

drop tablespace ts_history_24;
create regular tablespace ts_history_24 pagesize 16K
managed by database
using
(
  device '/dev/rD1F08V3HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_25 of D1

drop tablespace ts_history_25;
create regular tablespace ts_history_25 pagesize 16K
managed by database
using
(
  device '/dev/rD1F09V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_26 of D1

drop tablespace ts_history_26;
create regular tablespace ts_history_26 pagesize 16K
managed by database
using
(
  device '/dev/rD1F09V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_27 of D1

drop tablespace ts_history_27;
create regular tablespace ts_history_27 pagesize 16K
managed by database
using
(
  device '/dev/rD1F09V3HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_28 of D1

drop tablespace ts_history_28;
create regular tablespace ts_history_28 pagesize 16K
managed by database
using

```

```

(
  device '/dev/rD1F10V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_29 of D1

drop tablespace ts_history_29;
create regular tablespace ts_history_29 pagesize 16K
managed by database
using
(
  device '/dev/rD1F10V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_30 of D1

drop tablespace ts_history_30;
create regular tablespace ts_history_30 pagesize 16K
managed by database
using
(
  device '/dev/rD1F10V3HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_31 of D1

drop tablespace ts_history_31;
create regular tablespace ts_history_31 pagesize 16K
managed by database
using
(
  device '/dev/rD1F11V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_32 of D1

drop tablespace ts_history_32;
create regular tablespace ts_history_32 pagesize 16K
managed by database
using
(
  device '/dev/rD1F11V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_33 of D1

drop tablespace ts_history_33;
create regular tablespace ts_history_33 pagesize 16K
managed by database
using
(
  device '/dev/rD1F11V3HIST' 394496
)

```

```

)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_34 of D1

drop tablespace ts_history_34;
create regular tablespace ts_history_34 pagesize 16K
managed by database
using
(
device '/dev/rD1F12V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_35 of D1

drop tablespace ts_history_35;
create regular tablespace ts_history_35 pagesize 16K
managed by database
using
(
device '/dev/rD1F12V2HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_36 of D1

drop tablespace ts_history_36;
create regular tablespace ts_history_36 pagesize 16K
managed by database
using
(
device '/dev/rD1F12V3HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_37 of D1

drop tablespace ts_history_37;
create regular tablespace ts_history_37 pagesize 16K
managed by database
using
(
device '/dev/rD1F13V1HIST' 394496
)
extentsize 256
prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

-- now creating TS for ts_history_38 of D1

drop tablespace ts_history_38;
create regular tablespace ts_history_38 pagesize 16K
managed by database
using
(
device '/dev/rD1F13V2HIST' 394496
)
extentsize 256

```

```

prefetchsize 2048
bufferpool ibmdefaultbp16K;
commit;

connect reset;

ts/crts_item.ddl

connect to tpcc;

-- now creating TS for ts_item_01 of D1
drop tablespace ts_item_01;
create regular tablespace ts_item_01 pagesize 8K
managed by database
using
(
device '/dev/rD1F01V1ITEM' 64,
device '/dev/rD1F01V2ITEM' 64,
device '/dev/rD1F01V3ITEM' 64,
device '/dev/rD1F02V1ITEM' 64,
device '/dev/rD1F02V2ITEM' 64,
device '/dev/rD1F02V3ITEM' 64,
device '/dev/rD1F03V1ITEM' 64,
device '/dev/rD1F03V2ITEM' 64,
device '/dev/rD1F03V3ITEM' 64,
device '/dev/rD1F04V1ITEM' 64,
device '/dev/rD1F04V2ITEM' 64,
device '/dev/rD1F04V3ITEM' 64,
device '/dev/rD1F05V1ITEM' 64,
device '/dev/rD1F05V2ITEM' 64,
device '/dev/rD1F05V3ITEM' 64,
device '/dev/rD1F06V1ITEM' 64,
device '/dev/rD1F06V2ITEM' 64,
device '/dev/rD1F06V3ITEM' 64,
device '/dev/rD1F07V1ITEM' 64,
device '/dev/rD1F07V2ITEM' 64,
device '/dev/rD1F07V3ITEM' 64,
device '/dev/rD1F08V1ITEM' 64,
device '/dev/rD1F08V2ITEM' 64,
device '/dev/rD1F08V3ITEM' 64,
device '/dev/rD1F09V1ITEM' 64,
device '/dev/rD1F09V2ITEM' 64,
device '/dev/rD1F09V3ITEM' 64,
device '/dev/rD1F10V1ITEM' 64,
device '/dev/rD1F10V2ITEM' 64,
device '/dev/rD1F10V3ITEM' 64,
device '/dev/rD1F11V1ITEM' 64,
device '/dev/rD1F11V2ITEM' 64,
device '/dev/rD1F11V3ITEM' 64,
device '/dev/rD1F12V1ITEM' 64,
device '/dev/rD1F12V2ITEM' 64,
device '/dev/rD1F12V3ITEM' 64,
device '/dev/rD1F13V1ITEM' 64,
device '/dev/rD1F13V2ITEM' 64,
device '/dev/rD1F13V3ITEM' 64
)
extentsize 16
prefetchsize 2048
bufferpool ibmdefault8K;
commit;

connect reset;

ts/crts_neword.ddl

connect to tpcc;

-- now creating TS for ts_neword_01 of D1
drop tablespace ts_neword_01;

```

```

create regular tablespace ts_neword_01 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_02 of D1

drop tablespace ts_neword_02;
create regular tablespace ts_neword_02 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V2NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_03 of D1

drop tablespace ts_neword_03;
create regular tablespace ts_neword_03 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V3NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_04 of D1

drop tablespace ts_neword_04;
create regular tablespace ts_neword_04 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_05 of D1

drop tablespace ts_neword_05;
create regular tablespace ts_neword_05 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V2NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_06 of D1

drop tablespace ts_neword_06;
create regular tablespace ts_neword_06 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V3NOR' 219904
  )
  extentsize 256

```

```

  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_07 of D1

drop tablespace ts_neword_07;
create regular tablespace ts_neword_07 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_08 of D1

drop tablespace ts_neword_08;
create regular tablespace ts_neword_08 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V2NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_09 of D1

drop tablespace ts_neword_09;
create regular tablespace ts_neword_09 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_10 of D1

drop tablespace ts_neword_10;
create regular tablespace ts_neword_10 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_11 of D1

drop tablespace ts_neword_11;
create regular tablespace ts_neword_11 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V2NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_12 of D1

drop tablespace ts_neword_12;

```

```

create regular tablespace ts_neword_12 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V3NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_13 of D1

drop tablespace ts_neword_13;
create regular tablespace ts_neword_13 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_14 of D1

drop tablespace ts_neword_14;
create regular tablespace ts_neword_14 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V2NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_15 of D1

drop tablespace ts_neword_15;
create regular tablespace ts_neword_15 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V3NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_16 of D1

drop tablespace ts_neword_16;
create regular tablespace ts_neword_16 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_17 of D1

drop tablespace ts_neword_17;
create regular tablespace ts_neword_17 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V2NOR' 219904
  )
  extentsize 256

```

```

  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_18 of D1

drop tablespace ts_neword_18;
create regular tablespace ts_neword_18 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V3NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_19 of D1

drop tablespace ts_neword_19;
create regular tablespace ts_neword_19 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_20 of D1

drop tablespace ts_neword_20;
create regular tablespace ts_neword_20 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V2NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_21 of D1

drop tablespace ts_neword_21;
create regular tablespace ts_neword_21 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F07V3NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_22 of D1

drop tablespace ts_neword_22;
create regular tablespace ts_neword_22 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V1NOR' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_23 of D1

drop tablespace ts_neword_23;
create regular tablespace ts_neword_23 pagesize 4K

```



```

managed by database
using
(
  device '/dev/rD1F08V2NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_24 of D1

drop tablespace ts_neword_24;
create regular tablespace ts_neword_24 pagesize 4K
managed by database
using
(
  device '/dev/rD1F08V3NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_25 of D1

drop tablespace ts_neword_25;
create regular tablespace ts_neword_25 pagesize 4K
managed by database
using
(
  device '/dev/rD1F09V1NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_26 of D1

drop tablespace ts_neword_26;
create regular tablespace ts_neword_26 pagesize 4K
managed by database
using
(
  device '/dev/rD1F09V2NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_27 of D1

drop tablespace ts_neword_27;
create regular tablespace ts_neword_27 pagesize 4K
managed by database
using
(
  device '/dev/rD1F09V3NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_28 of D1

drop tablespace ts_neword_28;
create regular tablespace ts_neword_28 pagesize 4K
managed by database
using
(
  device '/dev/rD1F10V1NOR' 219904
)
extentsize 256
prefetchsize 2048;

```

```

commit;

-- now creating TS for ts_neword_29 of D1

drop tablespace ts_neword_29;
create regular tablespace ts_neword_29 pagesize 4K
managed by database
using
(
  device '/dev/rD1F10V2NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_30 of D1

drop tablespace ts_neword_30;
create regular tablespace ts_neword_30 pagesize 4K
managed by database
using
(
  device '/dev/rD1F10V3NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_31 of D1

drop tablespace ts_neword_31;
create regular tablespace ts_neword_31 pagesize 4K
managed by database
using
(
  device '/dev/rD1F11V1NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_32 of D1

drop tablespace ts_neword_32;
create regular tablespace ts_neword_32 pagesize 4K
managed by database
using
(
  device '/dev/rD1F11V2NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_33 of D1

drop tablespace ts_neword_33;
create regular tablespace ts_neword_33 pagesize 4K
managed by database
using
(
  device '/dev/rD1F11V3NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_34 of D1

drop tablespace ts_neword_34;
create regular tablespace ts_neword_34 pagesize 4K
managed by database

```

```

using
(
  device '/dev/rD1F12V1NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_35 of D1

drop tablespace ts_neword_35;
create regular tablespace ts_neword_35 pagesize 4K
managed by database
using
(
  device '/dev/rD1F12V2NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_36 of D1

drop tablespace ts_neword_36;
create regular tablespace ts_neword_36 pagesize 4K
managed by database
using
(
  device '/dev/rD1F12V3NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_37 of D1

drop tablespace ts_neword_37;
create regular tablespace ts_neword_37 pagesize 4K
managed by database
using
(
  device '/dev/rD1F13V1NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_38 of D1

drop tablespace ts_neword_38;
create regular tablespace ts_neword_38 pagesize 4K
managed by database
using
(
  device '/dev/rD1F13V2NOR' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_46 of D1

drop tablespace ts_neword_46;
create regular tablespace ts_neword_46 pagesize 4K
managed by database
using
(
  device '/dev/rD1F01V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

```

```

-- now creating TS for ts_neword_47 of D1

drop tablespace ts_neword_47;
create regular tablespace ts_neword_47 pagesize 4K
managed by database
using
(
  device '/dev/rD1F01V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_48 of D1

drop tablespace ts_neword_48;
create regular tablespace ts_neword_48 pagesize 4K
managed by database
using
(
  device '/dev/rD1F01V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_49 of D1

drop tablespace ts_neword_49;
create regular tablespace ts_neword_49 pagesize 4K
managed by database
using
(
  device '/dev/rD1F02V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_50 of D1

drop tablespace ts_neword_50;
create regular tablespace ts_neword_50 pagesize 4K
managed by database
using
(
  device '/dev/rD1F02V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_51 of D1

drop tablespace ts_neword_51;
create regular tablespace ts_neword_51 pagesize 4K
managed by database
using
(
  device '/dev/rD1F02V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_52 of D1

drop tablespace ts_neword_52;
create regular tablespace ts_neword_52 pagesize 4K
managed by database
using

```

```

(
  device '/dev/rD1F03V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_53 of D1

drop tablespace ts_neword_53;
create regular tablespace ts_neword_53 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V2NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_54 of D1

drop tablespace ts_neword_54;
create regular tablespace ts_neword_54 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_55 of D1

drop tablespace ts_neword_55;
create regular tablespace ts_neword_55 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_56 of D1

drop tablespace ts_neword_56;
create regular tablespace ts_neword_56 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V2NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_57 of D1

drop tablespace ts_neword_57;
create regular tablespace ts_neword_57 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V3NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

```

```

-- now creating TS for ts_neword_58 of D1

drop tablespace ts_neword_58;
create regular tablespace ts_neword_58 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V1NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_59 of D1

drop tablespace ts_neword_59;
create regular tablespace ts_neword_59 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V2NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_60 of D1

drop tablespace ts_neword_60;
create regular tablespace ts_neword_60 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F05V3NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_61 of D1

drop tablespace ts_neword_61;
create regular tablespace ts_neword_61 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V1NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_62 of D1

drop tablespace ts_neword_62;
create regular tablespace ts_neword_62 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F06V2NORI' 219904
  )
  extentsize 256
  prefetchsize 2048;
commit;

-- now creating TS for ts_neword_63 of D1

drop tablespace ts_neword_63;
create regular tablespace ts_neword_63 pagesize 4K
  managed by database
  using
  (

```

```

        device '/dev/rD1F06V3NORI' 219904
    )
    extentsize 256
    prefetchsize 2048;
commit;

-- now creating TS for ts_neword_64 of D1

drop tablespace ts_neword_64;
create regular tablespace ts_neword_64 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_65 of D1

drop tablespace ts_neword_65;
create regular tablespace ts_neword_65 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_66 of D1

drop tablespace ts_neword_66;
create regular tablespace ts_neword_66 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_67 of D1

drop tablespace ts_neword_67;
create regular tablespace ts_neword_67 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_68 of D1

drop tablespace ts_neword_68;
create regular tablespace ts_neword_68 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_69 of D1

```

```

drop tablespace ts_neword_69;
create regular tablespace ts_neword_69 pagesize 4K
managed by database
using
(
    device '/dev/rD1F08V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_70 of D1

drop tablespace ts_neword_70;
create regular tablespace ts_neword_70 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_71 of D1

drop tablespace ts_neword_71;
create regular tablespace ts_neword_71 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_72 of D1

drop tablespace ts_neword_72;
create regular tablespace ts_neword_72 pagesize 4K
managed by database
using
(
    device '/dev/rD1F09V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_73 of D1

drop tablespace ts_neword_73;
create regular tablespace ts_neword_73 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_74 of D1

drop tablespace ts_neword_74;
create regular tablespace ts_neword_74 pagesize 4K
managed by database
using
(
    device '/dev/rD1F10V2NORI' 219904
)

```

```

)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_75 of D1

drop tablespace ts_neword_75;
create regular tablespace ts_neword_75 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_76 of D1

drop tablespace ts_neword_76;
create regular tablespace ts_neword_76 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_77 of D1

drop tablespace ts_neword_77;
create regular tablespace ts_neword_77 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_78 of D1

drop tablespace ts_neword_78;
create regular tablespace ts_neword_78 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_79 of D1

drop tablespace ts_neword_79;
create regular tablespace ts_neword_79 pagesize 4K
managed by database
using
(
device '/dev/rD1F12V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_80 of D1

```

```

drop tablespace ts_neword_80;
create regular tablespace ts_neword_80 pagesize 4K
managed by database
using
(
device '/dev/rD1F12V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_81 of D1

drop tablespace ts_neword_81;
create regular tablespace ts_neword_81 pagesize 4K
managed by database
using
(
device '/dev/rD1F12V3NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_82 of D1

drop tablespace ts_neword_82;
create regular tablespace ts_neword_82 pagesize 4K
managed by database
using
(
device '/dev/rD1F13V1NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

-- now creating TS for ts_neword_83 of D1

drop tablespace ts_neword_83;
create regular tablespace ts_neword_83 pagesize 4K
managed by database
using
(
device '/dev/rD1F13V2NORI' 219904
)
extentsize 256
prefetchsize 2048;
commit;

connect reset;

ts/crts_order.ddl

connect to tpc;

-- now creating TS for ts_order_01 of D1

drop tablespace ts_order_01;
create regular tablespace ts_order_01 pagesize 8K
managed by database
using
(
device '/dev/rD1F01V1ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_02 of D1

```

```

drop tablespace ts_order_02;
create regular tablespace ts_order_02 pagesize 8K
managed by database
using
(
  device '/dev/rD1F01V2ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_03 of D1

```

drop tablespace ts_order_03;
create regular tablespace ts_order_03 pagesize 8K
managed by database
using
(
  device '/dev/rD1F01V3ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_04 of D1

```

drop tablespace ts_order_04;
create regular tablespace ts_order_04 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V1ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_05 of D1

```

drop tablespace ts_order_05;
create regular tablespace ts_order_05 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V2ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_06 of D1

```

drop tablespace ts_order_06;
create regular tablespace ts_order_06 pagesize 8K
managed by database
using
(
  device '/dev/rD1F02V3ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_07 of D1

```

drop tablespace ts_order_07;

```

```

create regular tablespace ts_order_07 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V1ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_08 of D1

```

drop tablespace ts_order_08;
create regular tablespace ts_order_08 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V2ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_09 of D1

```

drop tablespace ts_order_09;
create regular tablespace ts_order_09 pagesize 8K
managed by database
using
(
  device '/dev/rD1F03V3ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_10 of D1

```

drop tablespace ts_order_10;
create regular tablespace ts_order_10 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V1ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_11 of D1

```

drop tablespace ts_order_11;
create regular tablespace ts_order_11 pagesize 8K
managed by database
using
(
  device '/dev/rD1F04V2ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

-- now creating TS for ts_order_12 of D1

```

drop tablespace ts_order_12;
create regular tablespace ts_order_12 pagesize 8K
managed by database

```

```

using
(
  device '/dev/rD1F04V3ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_13 of D1

drop tablespace ts_order_13;
create regular tablespace ts_order_13 pagesize 8K
  managed by database

using
(
  device '/dev/rD1F05V1ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_14 of D1

drop tablespace ts_order_14;
create regular tablespace ts_order_14 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V2ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_15 of D1

drop tablespace ts_order_15;
create regular tablespace ts_order_15 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V3ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_16 of D1

drop tablespace ts_order_16;
create regular tablespace ts_order_16 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V1ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_17 of D1

drop tablespace ts_order_17;
create regular tablespace ts_order_17 pagesize 8K
  managed by database

```

```

using
(
  device '/dev/rD1F06V2ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_18 of D1

drop tablespace ts_order_18;
create regular tablespace ts_order_18 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V3ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_19 of D1

drop tablespace ts_order_19;
create regular tablespace ts_order_19 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V1ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_20 of D1

drop tablespace ts_order_20;
create regular tablespace ts_order_20 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V2ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_21 of D1

drop tablespace ts_order_21;
create regular tablespace ts_order_21 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V3ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_22 of D1

drop tablespace ts_order_22;
create regular tablespace ts_order_22 pagesize 8K
  managed by database

```

```

using
(
  device '/dev/rD1F08V1ORD' 664576
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_23 of D1

drop tablespace ts_order_23;
create regular tablespace ts_order_23 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V2ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_24 of D1

drop tablespace ts_order_24;
create regular tablespace ts_order_24 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V3ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_25 of D1

drop tablespace ts_order_25;
create regular tablespace ts_order_25 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V1ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_26 of D1

drop tablespace ts_order_26;
create regular tablespace ts_order_26 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V2ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_27 of D1

drop tablespace ts_order_27;
create regular tablespace ts_order_27 pagesize 8K
  managed by database
  using
  (

```

```

    device '/dev/rD1F09V3ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_28 of D1

drop tablespace ts_order_28;
create regular tablespace ts_order_28 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V1ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_29 of D1

drop tablespace ts_order_29;
create regular tablespace ts_order_29 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V2ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_30 of D1

drop tablespace ts_order_30;
create regular tablespace ts_order_30 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V3ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_31 of D1

drop tablespace ts_order_31;
create regular tablespace ts_order_31 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V1ORD' 664576
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_32 of D1

drop tablespace ts_order_32;
create regular tablespace ts_order_32 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V2ORD' 664576
  )

```



```

    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_33 of D1

drop tablespace ts_order_33;
create regular tablespace ts_order_33 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F11V3ORD' 664576
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_34 of D1

drop tablespace ts_order_34;
create regular tablespace ts_order_34 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F12V1ORD' 664576
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_35 of D1

drop tablespace ts_order_35;
create regular tablespace ts_order_35 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F12V2ORD' 664576
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_36 of D1

drop tablespace ts_order_36;
create regular tablespace ts_order_36 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F12V3ORD' 664576
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_37 of D1

drop tablespace ts_order_37;
create regular tablespace ts_order_37 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F13V1ORD' 664576
    )
    extentsize 256
    prefetchsize 4096

```

```

    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_order_38 of D1

drop tablespace ts_order_38;
create regular tablespace ts_order_38 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F13V2ORD' 664576
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

ts/crts_orderline.ddl

```

connect to tpcc;
-- for database D1

-- now creating TS for ts_orderline_01 of D1

drop tablespace ts_orderline_01;
create regular tablespace ts_orderline_01 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F01V1ORL01' 11343104
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_02 of D1

drop tablespace ts_orderline_02;
create regular tablespace ts_orderline_02 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F01V2ORL02' 11343104
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_03 of D1

drop tablespace ts_orderline_03;
create regular tablespace ts_orderline_03 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F01V3ORL03' 11343104
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_04 of D1

drop tablespace ts_orderline_04;

```

```

create regular tablespace ts_orderline_04 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V1ORL04' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_05 of D1

drop tablespace ts_orderline_05;
create regular tablespace ts_orderline_05 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V2ORL05' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_06 of D1

drop tablespace ts_orderline_06;
create regular tablespace ts_orderline_06 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V3ORL06' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_07 of D1

drop tablespace ts_orderline_07;
create regular tablespace ts_orderline_07 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V1ORL07' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_08 of D1

drop tablespace ts_orderline_08;
create regular tablespace ts_orderline_08 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V2ORL08' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_09 of D1

drop tablespace ts_orderline_09;
create regular tablespace ts_orderline_09 pagesize 8K
  managed by database

```

```

  using
  (
    device '/dev/rD1F03V3ORL09' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_10 of D1

drop tablespace ts_orderline_10;
create regular tablespace ts_orderline_10 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V1ORL10' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_11 of D1

drop tablespace ts_orderline_11;
create regular tablespace ts_orderline_11 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V2ORL11' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_12 of D1

drop tablespace ts_orderline_12;
create regular tablespace ts_orderline_12 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V3ORL12' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_13 of D1

drop tablespace ts_orderline_13;
create regular tablespace ts_orderline_13 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V1ORL13' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_14 of D1

drop tablespace ts_orderline_14;
create regular tablespace ts_orderline_14 pagesize 8K
  managed by database
  using

```

```

(
  device '/dev/rD1F05V2ORL14' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_15 of D1

drop tablespace ts_orderline_15;
create regular tablespace ts_orderline_15 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V3ORL15' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_16 of D1

drop tablespace ts_orderline_16;
create regular tablespace ts_orderline_16 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V1ORL16' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_17 of D1

drop tablespace ts_orderline_17;
create regular tablespace ts_orderline_17 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V2ORL17' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_18 of D1

drop tablespace ts_orderline_18;
create regular tablespace ts_orderline_18 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V3ORL18' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_19 of D1

drop tablespace ts_orderline_19;
create regular tablespace ts_orderline_19 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V1ORL19' 11343104

```

```

)
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_20 of D1

drop tablespace ts_orderline_20;
create regular tablespace ts_orderline_20 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V2ORL20' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_21 of D1

drop tablespace ts_orderline_21;
create regular tablespace ts_orderline_21 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V3ORL21' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_22 of D1

drop tablespace ts_orderline_22;
create regular tablespace ts_orderline_22 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V1ORL22' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_23 of D1

drop tablespace ts_orderline_23;
create regular tablespace ts_orderline_23 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V2ORL23' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_24 of D1

drop tablespace ts_orderline_24;
create regular tablespace ts_orderline_24 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V3ORL24' 11343104
  )
  extentsize 256

```

```

prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_25 of D1

drop tablespace ts_orderline_25;
create regular tablespace ts_orderline_25 pagesize 8K
managed by database
using
(
  device '/dev/rD1F09V1ORL25' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_26 of D1

drop tablespace ts_orderline_26;
create regular tablespace ts_orderline_26 pagesize 8K
managed by database
using
(
  device '/dev/rD1F09V2ORL26' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_27 of D1

drop tablespace ts_orderline_27;
create regular tablespace ts_orderline_27 pagesize 8K
managed by database
using
(
  device '/dev/rD1F09V3ORL27' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_28 of D1

drop tablespace ts_orderline_28;
create regular tablespace ts_orderline_28 pagesize 8K
managed by database
using
(
  device '/dev/rD1F10V1ORL28' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_29 of D1

drop tablespace ts_orderline_29;
create regular tablespace ts_orderline_29 pagesize 8K
managed by database
using
(
  device '/dev/rD1F10V2ORL29' 11343104
)
extentsize 256
prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_30 of D1

drop tablespace ts_orderline_30;
create regular tablespace ts_orderline_30 pagesize 8K
managed by database
using
(
  device '/dev/rD1F10V3ORL30' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_31 of D1

drop tablespace ts_orderline_31;
create regular tablespace ts_orderline_31 pagesize 8K
managed by database
using
(
  device '/dev/rD1F11V1ORL31' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_32 of D1

drop tablespace ts_orderline_32;
create regular tablespace ts_orderline_32 pagesize 8K
managed by database
using
(
  device '/dev/rD1F11V2ORL32' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_33 of D1

drop tablespace ts_orderline_33;
create regular tablespace ts_orderline_33 pagesize 8K
managed by database
using
(
  device '/dev/rD1F11V3ORL33' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_34 of D1

drop tablespace ts_orderline_34;
create regular tablespace ts_orderline_34 pagesize 8K
managed by database
using
(
  device '/dev/rD1F12V1ORL34' 11343104
)
extentsize 256
prefetchsize 4096
bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for ts_orderline_35 of D1

drop tablespace ts_orderline_35;
create regular tablespace ts_orderline_35 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V2ORL35' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_36 of D1

drop tablespace ts_orderline_36;
create regular tablespace ts_orderline_36 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V3ORL36' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_37 of D1

drop tablespace ts_orderline_37;
create regular tablespace ts_orderline_37 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F13V1ORL37' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_38 of D1

drop tablespace ts_orderline_38;
create regular tablespace ts_orderline_38 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F13V2ORL38' 11343104
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

connect reset;

ts/crts ordidx.ddl

connect to tpcc;

-- now creating TS for is_order_01 of D1

drop tablespace is_order_01;
create regular tablespace is_order_01 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F01V1ORDI' 358912

```

```

    )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_02 of D1

drop tablespace is_order_02;
create regular tablespace is_order_02 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F01V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_03 of D1

drop tablespace is_order_03;
create regular tablespace is_order_03 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F01V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_04 of D1

drop tablespace is_order_04;
create regular tablespace is_order_04 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_05 of D1

drop tablespace is_order_05;
create regular tablespace is_order_05 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_06 of D1

drop tablespace is_order_06;
create regular tablespace is_order_06 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F02V3ORDI' 358912
  )
  extentsize 256

```

```

prefetchsize 4096
bufferpool ibmdefaultbp8K;

commit;

-- now creating TS for is_order_07 of D1

drop tablespace is_order_07;
create regular tablespace is_order_07 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_08 of D1

drop tablespace is_order_08;
create regular tablespace is_order_08 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_09 of D1

drop tablespace is_order_09;
create regular tablespace is_order_09 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F03V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_10 of D1

drop tablespace is_order_10;
create regular tablespace is_order_10 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_11 of D1

drop tablespace is_order_11;
create regular tablespace is_order_11 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096

```

```

bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_12 of D1

drop tablespace is_order_12;
create regular tablespace is_order_12 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F04V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_13 of D1

drop tablespace is_order_13;
create regular tablespace is_order_13 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_14 of D1

drop tablespace is_order_14;
create regular tablespace is_order_14 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_15 of D1

drop tablespace is_order_15;
create regular tablespace is_order_15 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F05V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_16 of D1

drop tablespace is_order_16;
create regular tablespace is_order_16 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

```

```

-- now creating TS for is_order_17 of D1

drop tablespace is_order_17;
create regular tablespace is_order_17 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_18 of D1

drop tablespace is_order_18;
create regular tablespace is_order_18 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F06V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_19 of D1

drop tablespace is_order_19;
create regular tablespace is_order_19 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_20 of D1

drop tablespace is_order_20;
create regular tablespace is_order_20 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_21 of D1

drop tablespace is_order_21;
create regular tablespace is_order_21 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F07V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_22 of D1

```

```

drop tablespace is_order_22;
create regular tablespace is_order_22 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_23 of D1

drop tablespace is_order_23;
create regular tablespace is_order_23 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_24 of D1

drop tablespace is_order_24;
create regular tablespace is_order_24 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F08V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_25 of D1

drop tablespace is_order_25;
create regular tablespace is_order_25 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_26 of D1

drop tablespace is_order_26;
create regular tablespace is_order_26 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_27 of D1

drop tablespace is_order_27;

```

```

create regular tablespace is_order_27 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F09V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_28 of D1

drop tablespace is_order_28;
create regular tablespace is_order_28 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_29 of D1

drop tablespace is_order_29;
create regular tablespace is_order_29 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_30 of D1

drop tablespace is_order_30;
create regular tablespace is_order_30 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F10V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_31 of D1

drop tablespace is_order_31;
create regular tablespace is_order_31 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_32 of D1

drop tablespace is_order_32;
create regular tablespace is_order_32 pagesize 8K
  managed by database

```

```

  using
  (
    device '/dev/rD1F11V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_33 of D1

drop tablespace is_order_33;
create regular tablespace is_order_33 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F11V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_34 of D1

drop tablespace is_order_34;
create regular tablespace is_order_34 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V1ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_35 of D1

drop tablespace is_order_35;
create regular tablespace is_order_35 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V2ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_36 of D1

drop tablespace is_order_36;
create regular tablespace is_order_36 pagesize 8K
  managed by database
  using
  (
    device '/dev/rD1F12V3ORDI' 358912
  )
  extentsize 256
  prefetchsize 4096
  bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_37 of D1

drop tablespace is_order_37;
create regular tablespace is_order_37 pagesize 8K
  managed by database
  using
  (

```



```

        device '/dev/rD1F13V1ORDI' 358912
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_order_38 of D1

drop tablespace is_order_38;
create regular tablespace is_order_38 pagesize 8K
    managed by database
    using
    (
        device '/dev/rD1F13V2ORDI' 358912
    )
    extentsize 256
    prefetchsize 4096
    bufferpool ibmdefaultbp8K;
commit;

connect reset;

```

ts/crts_stock.ddl

```

-- now creating TS for ts_stock_01
connect to tpcc;

drop tablespace ts_stock01;
create regular tablespace ts_stock_01 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F01V1STK01' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_46

drop tablespace ts_stock46;
create regular tablespace ts_stock_46 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F01V1STK46' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_02

drop tablespace ts_stock02;
create regular tablespace ts_stock_02 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F01V2STK02' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_47

drop tablespace ts_stock47;
create regular tablespace ts_stock_47 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F01V2STK47' 9415936
    )
    extentsize 256
    prefetchsize 4096;

```

```

-- now creating TS for ts_stock_03

drop tablespace ts_stock03;
create regular tablespace ts_stock_03 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F01V3STK03' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_48

drop tablespace ts_stock48;
create regular tablespace ts_stock_48 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F01V3STK48' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_04

drop tablespace ts_stock04;
create regular tablespace ts_stock_04 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F02V1STK04' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_49

drop tablespace ts_stock49;
create regular tablespace ts_stock_49 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F02V1STK49' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_05

drop tablespace ts_stock05;
create regular tablespace ts_stock_05 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F02V2STK05' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_50

drop tablespace ts_stock50;
create regular tablespace ts_stock_50 pagesize 4K
    managed by database
    using
    (
        device '/dev/rD1F02V2STK50' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_06

drop tablespace ts_stock06;
create regular tablespace ts_stock_06 pagesize 4K
    managed by database
    using

```

```

(
  device '/dev/rD1F02V3STK06' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_51

drop tablespace ts_stock51;
create regular tablespace ts_stock_51 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V3STK51' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_07

drop tablespace ts_stock07;
create regular tablespace ts_stock_07 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V1STK07' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_52

drop tablespace ts_stock52;
create regular tablespace ts_stock_52 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V1STK52' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_08

drop tablespace ts_stock08;
create regular tablespace ts_stock_08 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V2STK08' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_53

drop tablespace ts_stock53;
create regular tablespace ts_stock_53 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V2STK53' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_09

drop tablespace ts_stock09;
create regular tablespace ts_stock_09 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3STK09' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_54

```

```

drop tablespace ts_stock54;
create regular tablespace ts_stock_54 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3STK54' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_10

drop tablespace ts_stock10;
create regular tablespace ts_stock_10 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1STK10' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_55

drop tablespace ts_stock55;
create regular tablespace ts_stock_55 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1STK55' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_11

drop tablespace ts_stock11;
create regular tablespace ts_stock_11 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V2STK11' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_56

drop tablespace ts_stock56;
create regular tablespace ts_stock_56 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V2STK56' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_12

drop tablespace ts_stock12;
create regular tablespace ts_stock_12 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V3STK12' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_57

drop tablespace ts_stock57;
create regular tablespace ts_stock_57 pagesize 4K
  managed by database
  using
  (

```

```

        device '/dev/rD1F04V3STK57' 9415936
    )
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_13

drop tablespace ts_stock13;
create regular tablespace ts_stock_13 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1STK13' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_58

drop tablespace ts_stock58;
create regular tablespace ts_stock_58 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V1STK58' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_14

drop tablespace ts_stock14;
create regular tablespace ts_stock_14 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2STK14' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_59

drop tablespace ts_stock59;
create regular tablespace ts_stock_59 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V2STK59' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_15

drop tablespace ts_stock15;
create regular tablespace ts_stock_15 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3STK15' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_60

drop tablespace ts_stock60;
create regular tablespace ts_stock_60 pagesize 4K
managed by database
using
(
    device '/dev/rD1F05V3STK60' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_16

```

```

drop tablespace ts_stock16;
create regular tablespace ts_stock_16 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1STK16' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_61

drop tablespace ts_stock61;
create regular tablespace ts_stock_61 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V1STK61' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_17

drop tablespace ts_stock17;
create regular tablespace ts_stock_17 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2STK17' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_62

drop tablespace ts_stock62;
create regular tablespace ts_stock_62 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V2STK62' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_18

drop tablespace ts_stock18;
create regular tablespace ts_stock_18 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3STK18' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_63

drop tablespace ts_stock63;
create regular tablespace ts_stock_63 pagesize 4K
managed by database
using
(
    device '/dev/rD1F06V3STK63' 9415936
)
    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_19

drop tablespace ts_stock19;
create regular tablespace ts_stock_19 pagesize 4K
managed by database
using
(
    device '/dev/rD1F07V1STK19' 9415936
)

```

```

)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_64

drop tablespace ts_stock64;
create regular tablespace ts_stock_64 pagesize 4K
managed by database
using
(
device '/dev/rD1F07V1STK64' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_20

drop tablespace ts_stock20;
create regular tablespace ts_stock_20 pagesize 4K
managed by database
using
(
device '/dev/rD1F07V2STK20' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_65

drop tablespace ts_stock65;
create regular tablespace ts_stock_65 pagesize 4K
managed by database
using
(
device '/dev/rD1F07V2STK65' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_21

drop tablespace ts_stock21;
create regular tablespace ts_stock_21 pagesize 4K
managed by database
using
(
device '/dev/rD1F07V3STK21' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_66

drop tablespace ts_stock66;
create regular tablespace ts_stock_66 pagesize 4K
managed by database
using
(
device '/dev/rD1F07V3STK66' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_22

drop tablespace ts_stock22;
create regular tablespace ts_stock_22 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V1STK22' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_67

```

```

drop tablespace ts_stock67;
create regular tablespace ts_stock_67 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V1STK67' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_23

drop tablespace ts_stock23;
create regular tablespace ts_stock_23 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V2STK23' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_68

drop tablespace ts_stock68;
create regular tablespace ts_stock_68 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V2STK68' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_24

drop tablespace ts_stock24;
create regular tablespace ts_stock_24 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V3STK24' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_69

drop tablespace ts_stock69;
create regular tablespace ts_stock_69 pagesize 4K
managed by database
using
(
device '/dev/rD1F08V3STK69' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_25

drop tablespace ts_stock25;
create regular tablespace ts_stock_25 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V1STK25' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_70

drop tablespace ts_stock70;
create regular tablespace ts_stock_70 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V1STK70' 9415936
)

```

```

)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_26

drop tablespace ts_stock26;
create regular tablespace ts_stock_26 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V2STK26' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_71

drop tablespace ts_stock71;
create regular tablespace ts_stock_71 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V2STK71' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_27

drop tablespace ts_stock27;
create regular tablespace ts_stock_27 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V3STK27' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_72

drop tablespace ts_stock72;
create regular tablespace ts_stock_72 pagesize 4K
managed by database
using
(
device '/dev/rD1F09V3STK72' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_28

drop tablespace ts_stock28;
create regular tablespace ts_stock_28 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V1STK28' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_73

drop tablespace ts_stock73;
create regular tablespace ts_stock_73 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V1STK73' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_29

drop tablespace ts_stock29;

```

```

create regular tablespace ts_stock_29 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V2STK29' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_74

drop tablespace ts_stock74;
create regular tablespace ts_stock_74 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V2STK74' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_30

drop tablespace ts_stock30;
create regular tablespace ts_stock_30 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V3STK30' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_75

drop tablespace ts_stock75;
create regular tablespace ts_stock_75 pagesize 4K
managed by database
using
(
device '/dev/rD1F10V3STK75' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_31

drop tablespace ts_stock31;
create regular tablespace ts_stock_31 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V1STK31' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_76

drop tablespace ts_stock76;
create regular tablespace ts_stock_76 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V1STK76' 9415936
)
extentsize 256
prefetchsize 4096;
-- now creating TS for ts_stock_32

drop tablespace ts_stock32;
create regular tablespace ts_stock_32 pagesize 4K
managed by database
using
(
device '/dev/rD1F11V2STK32' 9415936
)

```

```

    extentsize 256
    prefetchsize 4096;
-- now creating TS for ts_stock_77

drop tablespace ts_stock77;
create regular tablespace ts_stock_77 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F11V2STK77' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_33

drop tablespace ts_stock33;
create regular tablespace ts_stock_33 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F11V3STK33' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_78

drop tablespace ts_stock78;
create regular tablespace ts_stock_78 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F11V3STK78' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_34

drop tablespace ts_stock34;
create regular tablespace ts_stock_34 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F12V1STK34' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_79

drop tablespace ts_stock79;
create regular tablespace ts_stock_79 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F12V1STK79' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_35

drop tablespace ts_stock35;
create regular tablespace ts_stock_35 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F12V2STK35' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_80

drop tablespace ts_stock80;
create regular tablespace ts_stock_80 pagesize 4K

```

```

  managed by database
  using
  (
    device '/dev/rD1F12V2STK80' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_36

drop tablespace ts_stock36;
create regular tablespace ts_stock_36 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F12V3STK36' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_81

drop tablespace ts_stock81;
create regular tablespace ts_stock_81 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F12V3STK81' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_37

drop tablespace ts_stock37;
create regular tablespace ts_stock_37 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F13V1STK37' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_82

drop tablespace ts_stock82;
create regular tablespace ts_stock_82 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F13V1STK82' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_38

drop tablespace ts_stock38;
create regular tablespace ts_stock_38 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F13V2STK38' 9415936
  )
  extentsize 256
  prefetchsize 4096;
-- now creating TS for ts_stock_83

drop tablespace ts_stock83;
create regular tablespace ts_stock_83 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F13V2STK83' 9415936
  )

```

```

extentsize 256
prefetchsize 4096;

connect reset;

                                ts/crts_ware.ddl

connect to tpc;

drop tablespace ts_ware_01;
create regular tablespace ts_ware_01 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V1WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_02 of D1

drop tablespace ts_ware_02;
create regular tablespace ts_ware_02 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V2WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_03 of D1

drop tablespace ts_ware_03;
create regular tablespace ts_ware_03 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F01V3WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_04 of D1

drop tablespace ts_ware_04;
create regular tablespace ts_ware_04 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V1WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_05 of D1

drop tablespace ts_ware_05;
create regular tablespace ts_ware_05 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V2WARE' 192
  )
  extentsize 16
  prefetchsize 2048;

```

```

commit;

-- now creating TS for ts_ware_06 of D1

drop tablespace ts_ware_06;
create regular tablespace ts_ware_06 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F02V3WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_07 of D1

drop tablespace ts_ware_07;
create regular tablespace ts_ware_07 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V1WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_08 of D1

drop tablespace ts_ware_08;
create regular tablespace ts_ware_08 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V2WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_09 of D1

drop tablespace ts_ware_09;
create regular tablespace ts_ware_09 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F03V3WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_10 of D1

drop tablespace ts_ware_10;
create regular tablespace ts_ware_10 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F04V1WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_11 of D1

drop tablespace ts_ware_11;
create regular tablespace ts_ware_11 pagesize 4K
  managed by database

```

```

using
(
  device '/dev/rD1F04V2WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_12 of D1

drop tablespace ts_ware_12;
create regular tablespace ts_ware_12 pagesize 4K
managed by database
using
(
  device '/dev/rD1F04V3WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_13 of D1

drop tablespace ts_ware_13;
create regular tablespace ts_ware_13 pagesize 4K
managed by database
using
(
  device '/dev/rD1F05V1WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_14 of D1

drop tablespace ts_ware_14;
create regular tablespace ts_ware_14 pagesize 4K
managed by database
using
(
  device '/dev/rD1F05V2WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_15 of D1

drop tablespace ts_ware_15;
create regular tablespace ts_ware_15 pagesize 4K
managed by database
using
(
  device '/dev/rD1F05V3WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_16 of D1

drop tablespace ts_ware_16;
create regular tablespace ts_ware_16 pagesize 4K
managed by database
using
(
  device '/dev/rD1F06V1WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

```

```

-- now creating TS for ts_ware_17 of D1

drop tablespace ts_ware_17;
create regular tablespace ts_ware_17 pagesize 4K
managed by database
using
(
  device '/dev/rD1F06V2WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_18 of D1

drop tablespace ts_ware_18;
create regular tablespace ts_ware_18 pagesize 4K
managed by database
using
(
  device '/dev/rD1F06V3WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_19 of D1

drop tablespace ts_ware_19;
create regular tablespace ts_ware_19 pagesize 4K
managed by database
using
(
  device '/dev/rD1F07V1WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_20 of D1

drop tablespace ts_ware_20;
create regular tablespace ts_ware_20 pagesize 4K
managed by database
using
(
  device '/dev/rD1F07V2WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_21 of D1

drop tablespace ts_ware_21;
create regular tablespace ts_ware_21 pagesize 4K
managed by database
using
(
  device '/dev/rD1F07V3WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_22 of D1

drop tablespace ts_ware_22;
create regular tablespace ts_ware_22 pagesize 4K
managed by database
using

```



```

(
  device '/dev/rD1F08V1WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_23 of D1

drop tablespace ts_ware_23;
create regular tablespace ts_ware_23 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V2WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_24 of D1

drop tablespace ts_ware_24;
create regular tablespace ts_ware_24 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F08V3WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_25 of D1

drop tablespace ts_ware_25;
create regular tablespace ts_ware_25 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V1WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_26 of D1

drop tablespace ts_ware_26;
create regular tablespace ts_ware_26 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V2WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_27 of D1

drop tablespace ts_ware_27;
create regular tablespace ts_ware_27 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F09V3WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

```

```

-- now creating TS for ts_ware_28 of D1

drop tablespace ts_ware_28;
create regular tablespace ts_ware_28 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V1WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_29 of D1

drop tablespace ts_ware_29;
create regular tablespace ts_ware_29 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V2WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_30 of D1

drop tablespace ts_ware_30;
create regular tablespace ts_ware_30 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F10V3WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_31 of D1

drop tablespace ts_ware_31;
create regular tablespace ts_ware_31 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F11V1WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_32 of D1

drop tablespace ts_ware_32;
create regular tablespace ts_ware_32 pagesize 4K
  managed by database
  using
  (
    device '/dev/rD1F11V2WARE' 192
  )
  extentsize 16
  prefetchsize 2048;
commit;

-- now creating TS for ts_ware_33 of D1

drop tablespace ts_ware_33;
create regular tablespace ts_ware_33 pagesize 4K
  managed by database
  using
  (

```

```

device '/dev/rD1F11V3WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_34 of D1

drop tablespace ts_ware_34;
create regular tablespace ts_ware_34 pagesize 4K
managed by database
using
(
device '/dev/rD1F12V1WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_35 of D1

drop tablespace ts_ware_35;
create regular tablespace ts_ware_35 pagesize 4K
managed by database
using
(
device '/dev/rD1F12V2WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_36 of D1

drop tablespace ts_ware_36;
create regular tablespace ts_ware_36 pagesize 4K
managed by database
using
(
device '/dev/rD1F12V3WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_37 of D1

drop tablespace ts_ware_37;
create regular tablespace ts_ware_37 pagesize 4K
managed by database
using
(
device '/dev/rD1F13V1WARE' 192
)
extentsize 16
prefetchsize 2048;
commit;

-- now creating TS for ts_ware_38 of D1

drop tablespace ts_ware_38;
create regular tablespace ts_ware_38 pagesize 4K
managed by database
using
(
device '/dev/rD1F13V2WARE' 192
)
extentsize 16
prefetchsize 2048;
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;

-- Non-Affinitized Bufferpools

-- ITM: 1,500 pages in total
alter bufferpool ITM DEFERRED size 2000;

-- WAR: 4,000 pages in total
alter bufferpool WAR1 DEFERRED size 950;
alter bufferpool WAR2 DEFERRED size 1050;
alter bufferpool WAR3 DEFERRED size 1050;
alter bufferpool WAR4 DEFERRED size 950;

-- DIS: 40,000 pages in total
alter bufferpool DIS1 DEFERRED size 9500;
alter bufferpool DIS2 DEFERRED size 10500;
alter bufferpool DIS3 DEFERRED size 10500;
alter bufferpool DIS4 DEFERRED size 9500;

-- HST: 2,500 pages in total
alter bufferpool HST1 DEFERRED size 590;
alter bufferpool HST2 DEFERRED size 660;
alter bufferpool HST3 DEFERRED size 660;
alter bufferpool HST4 DEFERRED size 590;

-- NEW: 1,850,000 pages in total
alter bufferpool NEW1 DEFERRED size 438450;
alter bufferpool NEW2 DEFERRED size 486550;
alter bufferpool NEW3 DEFERRED size 486550;
alter bufferpool NEW4 DEFERRED size 438450;

-- ORD: 1,900,000 pages in total
alter bufferpool ORD1 DEFERRED size 450300;
alter bufferpool ORD2 DEFERRED size 499700;
alter bufferpool ORD3 DEFERRED size 499700;
alter bufferpool ORD4 DEFERRED size 450300;

-- ORD_I: 6,000,000 pages in total
alter bufferpool ORD_I1 DEFERRED size 1422000;
alter bufferpool ORD_I2 DEFERRED size 1578000;
alter bufferpool ORD_I3 DEFERRED size 1578000;
alter bufferpool ORD_I4 DEFERRED size 1422000;

-- OLN: 9,500,000 pages in total
alter bufferpool OLN1 DEFERRED size 2251500;
alter bufferpool OLN2 DEFERRED size 2498500;
alter bufferpool OLN3 DEFERRED size 2498500;
alter bufferpool OLN4 DEFERRED size 2251500;

-- CST: 15,000,000 pages in total
alter bufferpool CST1 DEFERRED size 3555000;

```

```

alter bufferpool CST2 DEFERRED size 3945000;
alter bufferpool CST3 DEFERRED size 3945000;
alter bufferpool CST4 DEFERRED size 3555000;

-- CST_I: 5,000,000 pages in total
alter bufferpool CST_I1 DEFERRED size 1185000;
alter bufferpool CST_I2 DEFERRED size 1315000;
alter bufferpool CST_I3 DEFERRED size 1315000;
alter bufferpool CST_I4 DEFERRED size 1185000;

-- STK: 167,600,000 pages in total
alter bufferpool STK1 DEFERRED size 39720000;
alter bufferpool STK2 DEFERRED size 44080000;
alter bufferpool STK3 DEFERRED size 44080000;
alter bufferpool STK4 DEFERRED size 39720000;

connect reset;
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 tpc;

```

```

create bufferpool IBMDEFAULT8K size 1000 pagesize 8192;
create bufferpool IBMDEFAULT16K size 1000 pagesize 16384;

```

```

create bufferpool WAR1 size 250 pagesize 4096;
create bufferpool WAR2 size 250 pagesize 4096;
create bufferpool WAR3 size 250 pagesize 4096;
create bufferpool WAR4 size 250 pagesize 4096;

```

```

create bufferpool DIS1 size 250 pagesize 4096;
create bufferpool DIS2 size 250 pagesize 4096;
create bufferpool DIS3 size 250 pagesize 4096;
create bufferpool DIS4 size 250 pagesize 4096;

```

```

create bufferpool ITM size 1000 pagesize 8192;

```

```

create bufferpool HST1 size 250 pagesize 16384;
create bufferpool HST2 size 250 pagesize 16384;
create bufferpool HST3 size 250 pagesize 16384;
create bufferpool HST4 size 250 pagesize 16384;

```

```

create bufferpool NEW1 size 250 pagesize 4096;
create bufferpool NEW2 size 250 pagesize 4096;
create bufferpool NEW3 size 250 pagesize 4096;
create bufferpool NEW4 size 250 pagesize 4096;

```

```

create bufferpool ORD1 size 250 pagesize 8192;
create bufferpool ORD2 size 250 pagesize 8192;
create bufferpool ORD3 size 250 pagesize 8192;
create bufferpool ORD4 size 250 pagesize 8192;

```

```

create bufferpool ORD_I1 size 250 pagesize 8192;
create bufferpool ORD_I2 size 250 pagesize 8192;
create bufferpool ORD_I3 size 250 pagesize 8192;
create bufferpool ORD_I4 size 250 pagesize 8192;

```

```

create bufferpool OLN1 size 250 pagesize 8192;
create bufferpool OLN2 size 250 pagesize 8192;
create bufferpool OLN3 size 250 pagesize 8192;
create bufferpool OLN4 size 250 pagesize 8192;

```

```

create bufferpool CST1 size 250 pagesize 4096;
create bufferpool CST2 size 250 pagesize 4096;
create bufferpool CST3 size 250 pagesize 4096;
create bufferpool CST4 size 250 pagesize 4096;

```

```

create bufferpool CST_I1 size 250 pagesize 8192;
create bufferpool CST_I2 size 250 pagesize 8192;
create bufferpool CST_I3 size 250 pagesize 8192;
create bufferpool CST_I4 size 250 pagesize 8192;

```

```

create bufferpool STK1 size 250 pagesize 4096;
create bufferpool STK2 size 250 pagesize 4096;
create bufferpool STK3 size 250 pagesize 4096;
create bufferpool STK4 size 250 pagesize 4096;

```

```

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 tpc;

```

```

-----
-- Non-Affinitized Bufferpools
-----

```

```

alter tablespace TS_ITEM_01 bufferpool ITM;

```

```

-----
-- Affinitized Bufferpools
-----

```

```

alter tablespace TS_WARE_01 bufferpool WAR1;
alter tablespace TS_WARE_02 bufferpool WAR1;
alter tablespace TS_WARE_03 bufferpool WAR1;
alter tablespace TS_WARE_04 bufferpool WAR1;
alter tablespace TS_WARE_05 bufferpool WAR1;
alter tablespace TS_WARE_06 bufferpool WAR1;
alter tablespace TS_WARE_07 bufferpool WAR1;
alter tablespace TS_WARE_08 bufferpool WAR1;
alter tablespace TS_WARE_09 bufferpool WAR1;
alter tablespace TS_WARE_10 bufferpool WAR2;
alter tablespace TS_WARE_11 bufferpool WAR2;
alter tablespace TS_WARE_12 bufferpool WAR2;
alter tablespace TS_WARE_13 bufferpool WAR2;
alter tablespace TS_WARE_14 bufferpool WAR2;
alter tablespace TS_WARE_15 bufferpool WAR2;
alter tablespace TS_WARE_16 bufferpool WAR2;
alter tablespace TS_WARE_17 bufferpool WAR2;
alter tablespace TS_WARE_18 bufferpool WAR2;
alter tablespace TS_WARE_19 bufferpool WAR2;
alter tablespace TS_WARE_20 bufferpool WAR3;

```



```

alter tablespace ts_stock_71 prefetchsize 4096;
alter tablespace ts_stock_72 prefetchsize 4096;
alter tablespace ts_stock_73 prefetchsize 4096;
alter tablespace ts_stock_74 prefetchsize 4096;
alter tablespace ts_stock_75 prefetchsize 4096;
alter tablespace ts_stock_76 prefetchsize 4096;
alter tablespace ts_ware_01 prefetchsize 4096;
alter tablespace ts_ware_02 prefetchsize 4096;
alter tablespace ts_ware_03 prefetchsize 4096;
alter tablespace ts_ware_04 prefetchsize 4096;
alter tablespace ts_ware_05 prefetchsize 4096;
alter tablespace ts_ware_06 prefetchsize 4096;
alter tablespace ts_ware_07 prefetchsize 4096;
alter tablespace ts_ware_08 prefetchsize 4096;
alter tablespace ts_ware_09 prefetchsize 4096;
alter tablespace ts_ware_10 prefetchsize 4096;
alter tablespace ts_ware_11 prefetchsize 4096;
alter tablespace ts_ware_12 prefetchsize 4096;
alter tablespace ts_ware_13 prefetchsize 4096;
alter tablespace ts_ware_14 prefetchsize 4096;
alter tablespace ts_ware_15 prefetchsize 4096;
alter tablespace ts_ware_16 prefetchsize 4096;
alter tablespace ts_ware_17 prefetchsize 4096;
alter tablespace ts_ware_18 prefetchsize 4096;
alter tablespace ts_ware_19 prefetchsize 4096;
alter tablespace ts_ware_20 prefetchsize 4096;
alter tablespace ts_ware_21 prefetchsize 4096;
alter tablespace ts_ware_22 prefetchsize 4096;
alter tablespace ts_ware_23 prefetchsize 4096;
alter tablespace ts_ware_24 prefetchsize 4096;
alter tablespace ts_ware_25 prefetchsize 4096;
alter tablespace ts_ware_26 prefetchsize 4096;
alter tablespace ts_ware_27 prefetchsize 4096;
alter tablespace ts_ware_28 prefetchsize 4096;
alter tablespace ts_ware_29 prefetchsize 4096;
alter tablespace ts_ware_30 prefetchsize 4096;
alter tablespace ts_ware_31 prefetchsize 4096;
alter tablespace ts_ware_32 prefetchsize 4096;
alter tablespace ts_ware_33 prefetchsize 4096;
alter tablespace ts_ware_34 prefetchsize 4096;
alter tablespace ts_ware_35 prefetchsize 4096;
alter tablespace ts_ware_36 prefetchsize 4096;
alter tablespace ts_ware_37 prefetchsize 4096;
alter tablespace ts_ware_38 prefetchsize 4096;
connect reset;

```

CRCONST CUSTOMER.ddl

```

connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER1 OFF;
ALTER TABLE CUSTOMER1 DROP CONSTRAINT CUSTOMER1CKC;
ALTER TABLE CUSTOMER1 ADD CONSTRAINT CUSTOMER1CKC
CHECK (C_W_ID BETWEEN 1 AND 2150);
SET INTEGRITY FOR CUSTOMER1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER2 OFF;
ALTER TABLE CUSTOMER2 DROP CONSTRAINT CUSTOMER2CKC;
ALTER TABLE CUSTOMER2 ADD CONSTRAINT CUSTOMER2CKC
CHECK (C_W_ID BETWEEN 2151 AND 4300);
SET INTEGRITY FOR CUSTOMER2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER3 OFF;
ALTER TABLE CUSTOMER3 DROP CONSTRAINT CUSTOMER3CKC;
ALTER TABLE CUSTOMER3 ADD CONSTRAINT CUSTOMER3CKC
CHECK (C_W_ID BETWEEN 4301 AND 6450);
SET INTEGRITY FOR CUSTOMER3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER4 OFF;

```

```

ALTER TABLE CUSTOMER4 DROP CONSTRAINT CUSTOMER4CKC;
ALTER TABLE CUSTOMER4 ADD CONSTRAINT CUSTOMER4CKC
CHECK (C_W_ID BETWEEN 6451 AND 8600);
SET INTEGRITY FOR CUSTOMER4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER5 OFF;
ALTER TABLE CUSTOMER5 DROP CONSTRAINT CUSTOMER5CKC;
ALTER TABLE CUSTOMER5 ADD CONSTRAINT CUSTOMER5CKC
CHECK (C_W_ID BETWEEN 8601 AND 10750);
SET INTEGRITY FOR CUSTOMER5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER6 OFF;
ALTER TABLE CUSTOMER6 DROP CONSTRAINT CUSTOMER6CKC;
ALTER TABLE CUSTOMER6 ADD CONSTRAINT CUSTOMER6CKC
CHECK (C_W_ID BETWEEN 10751 AND 12900);
SET INTEGRITY FOR CUSTOMER6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER7 OFF;
ALTER TABLE CUSTOMER7 DROP CONSTRAINT CUSTOMER7CKC;
ALTER TABLE CUSTOMER7 ADD CONSTRAINT CUSTOMER7CKC
CHECK (C_W_ID BETWEEN 12901 AND 15050);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER8 OFF;
ALTER TABLE CUSTOMER8 DROP CONSTRAINT CUSTOMER8CKC;
ALTER TABLE CUSTOMER8 ADD CONSTRAINT CUSTOMER8CKC
CHECK (C_W_ID BETWEEN 15051 AND 17200);
SET INTEGRITY FOR CUSTOMER8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER9 OFF;
ALTER TABLE CUSTOMER9 DROP CONSTRAINT CUSTOMER9CKC;
ALTER TABLE CUSTOMER9 ADD CONSTRAINT CUSTOMER9CKC
CHECK (C_W_ID BETWEEN 17201 AND 19350);
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 19351 AND 21500);
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 21501 AND 23650);
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 23651 AND 25800);
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 25801 AND 27950);
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 27951 AND 30100);
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 30101 AND 32250);
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 32251 AND 34400);
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 34401 AND 36550);
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 36551 AND 38700);
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 38701 AND 40850);
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 40851 AND 43000);
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 43001 AND 45150);
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 45151 AND 47300);
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 47301 AND 49450);
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 49451 AND 51600);

```

```

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 51601 AND 53750);
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 53751 AND 55900);
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 55901 AND 58050);
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 58051 AND 60200);
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 60201 AND 62350);
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 62351 AND 64500);
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 64501 AND 66650);
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 66651 AND 68800);
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 68801 AND 70950);
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 70951 AND 73100);
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 73101 AND 75250);
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 75251 AND 77400);
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 77401 AND 79550);
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 >= 79551);
SET INTEGRITY FOR CUSTOMER38 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 2150);
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 2151 AND 4300);
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 4301 AND 6450);
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 6451 AND 8600);
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 8601 AND 10750);
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 10751 AND 12900);

```

```

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 12901 AND 15050);
SET INTEGRITY FOR DISTRICT7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT8 OFF;
ALTER TABLE DISTRICT8 DROP CONSTRAINT DISTRICT8CKC;
ALTER TABLE DISTRICT8 ADD CONSTRAINT DISTRICT8CKC CHECK
(D_W_ID BETWEEN 15051 AND 17200);
SET INTEGRITY FOR DISTRICT8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT9 OFF;
ALTER TABLE DISTRICT9 DROP CONSTRAINT DISTRICT9CKC;
ALTER TABLE DISTRICT9 ADD CONSTRAINT DISTRICT9CKC CHECK
(D_W_ID BETWEEN 17201 AND 19350);
SET INTEGRITY FOR DISTRICT9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT10 OFF;
ALTER TABLE DISTRICT10 DROP CONSTRAINT DISTRICT10CKC;
ALTER TABLE DISTRICT10 ADD CONSTRAINT DISTRICT10CKC
CHECK (D_W_ID BETWEEN 19351 AND 21500);
SET INTEGRITY FOR DISTRICT10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT11 OFF;
ALTER TABLE DISTRICT11 DROP CONSTRAINT DISTRICT11CKC;
ALTER TABLE DISTRICT11 ADD CONSTRAINT DISTRICT11CKC
CHECK (D_W_ID BETWEEN 21501 AND 23650);
SET INTEGRITY FOR DISTRICT11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT12 OFF;
ALTER TABLE DISTRICT12 DROP CONSTRAINT DISTRICT12CKC;
ALTER TABLE DISTRICT12 ADD CONSTRAINT DISTRICT12CKC
CHECK (D_W_ID BETWEEN 23651 AND 25800);
SET INTEGRITY FOR DISTRICT12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT13 OFF;
ALTER TABLE DISTRICT13 DROP CONSTRAINT DISTRICT13CKC;
ALTER TABLE DISTRICT13 ADD CONSTRAINT DISTRICT13CKC
CHECK (D_W_ID BETWEEN 25801 AND 27950);
SET INTEGRITY FOR DISTRICT13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT14 OFF;
ALTER TABLE DISTRICT14 DROP CONSTRAINT DISTRICT14CKC;
ALTER TABLE DISTRICT14 ADD CONSTRAINT DISTRICT14CKC
CHECK (D_W_ID BETWEEN 27951 AND 30100);
SET INTEGRITY FOR DISTRICT14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT15 OFF;
ALTER TABLE DISTRICT15 DROP CONSTRAINT DISTRICT15CKC;
ALTER TABLE DISTRICT15 ADD CONSTRAINT DISTRICT15CKC
CHECK (D_W_ID BETWEEN 30101 AND 32250);
SET INTEGRITY FOR DISTRICT15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT16 OFF;
ALTER TABLE DISTRICT16 DROP CONSTRAINT DISTRICT16CKC;
ALTER TABLE DISTRICT16 ADD CONSTRAINT DISTRICT16CKC
CHECK (D_W_ID BETWEEN 32251 AND 34400);
SET INTEGRITY FOR DISTRICT16 ALL IMMEDIATE UNCHECKED;
connect reset;

```

```

connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT17 OFF;
ALTER TABLE DISTRICT17 DROP CONSTRAINT DISTRICT17CKC;
ALTER TABLE DISTRICT17 ADD CONSTRAINT DISTRICT17CKC
CHECK (D_W_ID BETWEEN 34401 AND 36550);
SET INTEGRITY FOR DISTRICT17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT18 OFF;
ALTER TABLE DISTRICT18 DROP CONSTRAINT DISTRICT18CKC;
ALTER TABLE DISTRICT18 ADD CONSTRAINT DISTRICT18CKC
CHECK (D_W_ID BETWEEN 36551 AND 38700);
SET INTEGRITY FOR DISTRICT18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT19 OFF;
ALTER TABLE DISTRICT19 DROP CONSTRAINT DISTRICT19CKC;
ALTER TABLE DISTRICT19 ADD CONSTRAINT DISTRICT19CKC
CHECK (D_W_ID BETWEEN 38701 AND 40850);
SET INTEGRITY FOR DISTRICT19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT20 OFF;
ALTER TABLE DISTRICT20 DROP CONSTRAINT DISTRICT20CKC;
ALTER TABLE DISTRICT20 ADD CONSTRAINT DISTRICT20CKC
CHECK (D_W_ID BETWEEN 40851 AND 43000);
SET INTEGRITY FOR DISTRICT20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT21 OFF;
ALTER TABLE DISTRICT21 DROP CONSTRAINT DISTRICT21CKC;
ALTER TABLE DISTRICT21 ADD CONSTRAINT DISTRICT21CKC
CHECK (D_W_ID BETWEEN 43001 AND 45150);
SET INTEGRITY FOR DISTRICT21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT22 OFF;
ALTER TABLE DISTRICT22 DROP CONSTRAINT DISTRICT22CKC;
ALTER TABLE DISTRICT22 ADD CONSTRAINT DISTRICT22CKC
CHECK (D_W_ID BETWEEN 45151 AND 47300);
SET INTEGRITY FOR DISTRICT22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT23 OFF;
ALTER TABLE DISTRICT23 DROP CONSTRAINT DISTRICT23CKC;
ALTER TABLE DISTRICT23 ADD CONSTRAINT DISTRICT23CKC
CHECK (D_W_ID BETWEEN 47301 AND 49450);
SET INTEGRITY FOR DISTRICT23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT24 OFF;
ALTER TABLE DISTRICT24 DROP CONSTRAINT DISTRICT24CKC;
ALTER TABLE DISTRICT24 ADD CONSTRAINT DISTRICT24CKC
CHECK (D_W_ID BETWEEN 49451 AND 51600);
SET INTEGRITY FOR DISTRICT24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT25 OFF;
ALTER TABLE DISTRICT25 DROP CONSTRAINT DISTRICT25CKC;
ALTER TABLE DISTRICT25 ADD CONSTRAINT DISTRICT25CKC
CHECK (D_W_ID BETWEEN 51601 AND 53750);
SET INTEGRITY FOR DISTRICT25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT26 OFF;
ALTER TABLE DISTRICT26 DROP CONSTRAINT DISTRICT26CKC;
ALTER TABLE DISTRICT26 ADD CONSTRAINT DISTRICT26CKC
CHECK (D_W_ID BETWEEN 53751 AND 55900);
SET INTEGRITY FOR DISTRICT26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT27 OFF;

```

```

ALTER TABLE DISTRICT27 DROP CONSTRAINT DISTRICT27CKC;
ALTER TABLE DISTRICT27 ADD CONSTRAINT DISTRICT27CKC
CHECK (D_W_ID BETWEEN 55901 AND 58050);
SET INTEGRITY FOR DISTRICT27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT28 OFF;
ALTER TABLE DISTRICT28 DROP CONSTRAINT DISTRICT28CKC;
ALTER TABLE DISTRICT28 ADD CONSTRAINT DISTRICT28CKC
CHECK (D_W_ID BETWEEN 58051 AND 60200);
SET INTEGRITY FOR DISTRICT28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT29 OFF;
ALTER TABLE DISTRICT29 DROP CONSTRAINT DISTRICT29CKC;
ALTER TABLE DISTRICT29 ADD CONSTRAINT DISTRICT29CKC
CHECK (D_W_ID BETWEEN 60201 AND 62350);
SET INTEGRITY FOR DISTRICT29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT30 OFF;
ALTER TABLE DISTRICT30 DROP CONSTRAINT DISTRICT30CKC;
ALTER TABLE DISTRICT30 ADD CONSTRAINT DISTRICT30CKC
CHECK (D_W_ID BETWEEN 62351 AND 64500);
SET INTEGRITY FOR DISTRICT30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT31 OFF;
ALTER TABLE DISTRICT31 DROP CONSTRAINT DISTRICT31CKC;
ALTER TABLE DISTRICT31 ADD CONSTRAINT DISTRICT31CKC
CHECK (D_W_ID BETWEEN 64501 AND 66650);
SET INTEGRITY FOR DISTRICT31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT32 OFF;
ALTER TABLE DISTRICT32 DROP CONSTRAINT DISTRICT32CKC;
ALTER TABLE DISTRICT32 ADD CONSTRAINT DISTRICT32CKC
CHECK (D_W_ID BETWEEN 66651 AND 68800);
SET INTEGRITY FOR DISTRICT32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT33 OFF;
ALTER TABLE DISTRICT33 DROP CONSTRAINT DISTRICT33CKC;
ALTER TABLE DISTRICT33 ADD CONSTRAINT DISTRICT33CKC
CHECK (D_W_ID BETWEEN 68801 AND 70950);
SET INTEGRITY FOR DISTRICT33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT34 OFF;
ALTER TABLE DISTRICT34 DROP CONSTRAINT DISTRICT34CKC;
ALTER TABLE DISTRICT34 ADD CONSTRAINT DISTRICT34CKC
CHECK (D_W_ID BETWEEN 70951 AND 73100);
SET INTEGRITY FOR DISTRICT34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT35 OFF;
ALTER TABLE DISTRICT35 DROP CONSTRAINT DISTRICT35CKC;
ALTER TABLE DISTRICT35 ADD CONSTRAINT DISTRICT35CKC
CHECK (D_W_ID BETWEEN 73101 AND 75250);
SET INTEGRITY FOR DISTRICT35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT36 OFF;
ALTER TABLE DISTRICT36 DROP CONSTRAINT DISTRICT36CKC;
ALTER TABLE DISTRICT36 ADD CONSTRAINT DISTRICT36CKC
CHECK (D_W_ID BETWEEN 75251 AND 77400);
SET INTEGRITY FOR DISTRICT36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT37 OFF;
ALTER TABLE DISTRICT37 DROP CONSTRAINT DISTRICT37CKC;

```



```
ALTER TABLE DISTRICT37 ADD CONSTRAINT DISTRICT37CKC
CHECK (D_W_ID BETWEEN 77401 AND 79550);
SET INTEGRITY FOR DISTRICT37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR DISTRICT38 OFF;
ALTER TABLE DISTRICT38 DROP CONSTRAINT DISTRICT38CKC;
ALTER TABLE DISTRICT38 ADD CONSTRAINT DISTRICT38CKC
CHECK (D_W_ID >= 79551);
SET INTEGRITY FOR DISTRICT38 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 2150);
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 2151 AND 4300);
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 4301 AND 6450);
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 6451 AND 8600);
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 8601 AND 10750);
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 10751 AND 12900);
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 12901 AND 15050);
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 15051 AND 17200);
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 17201 AND 19350);
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 19351 AND 21500);
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 21501 AND 23650);
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 23651 AND 25800);
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 25801 AND 27950);
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 27951 AND 30100);
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 30101 AND 32250);
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 BETWEEN 32251 AND 34400);
SET INTEGRITY FOR HISTORY16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY17 OFF;
ALTER TABLE HISTORY17 DROP CONSTRAINT HISTORY17CKC;
ALTER TABLE HISTORY17 ADD CONSTRAINT HISTORY17CKC
CHECK (H_W_ID BETWEEN 34401 AND 36550);
SET INTEGRITY FOR HISTORY17 ALL IMMEDIATE UNCHECKED;

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY18 OFF;
ALTER TABLE HISTORY18 DROP CONSTRAINT HISTORY18CKC;
ALTER TABLE HISTORY18 ADD CONSTRAINT HISTORY18CKC
CHECK (H_W_ID BETWEEN 36551 AND 38700);
SET INTEGRITY FOR HISTORY18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
```

```

SET INTEGRITY FOR HISTORY19 OFF;
ALTER TABLE HISTORY19 DROP CONSTRAINT HISTORY19CKC;
ALTER TABLE HISTORY19 ADD CONSTRAINT HISTORY19CKC
CHECK (H_W_ID BETWEEN 38701 AND 40850);
SET INTEGRITY FOR HISTORY19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY20 OFF;
ALTER TABLE HISTORY20 DROP CONSTRAINT HISTORY20CKC;
ALTER TABLE HISTORY20 ADD CONSTRAINT HISTORY20CKC
CHECK (H_W_ID BETWEEN 40851 AND 43000);
SET INTEGRITY FOR HISTORY20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY21 OFF;
ALTER TABLE HISTORY21 DROP CONSTRAINT HISTORY21CKC;
ALTER TABLE HISTORY21 ADD CONSTRAINT HISTORY21CKC
CHECK (H_W_ID BETWEEN 43001 AND 45150);
SET INTEGRITY FOR HISTORY21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY22 OFF;
ALTER TABLE HISTORY22 DROP CONSTRAINT HISTORY22CKC;
ALTER TABLE HISTORY22 ADD CONSTRAINT HISTORY22CKC
CHECK (H_W_ID BETWEEN 45151 AND 47300);
SET INTEGRITY FOR HISTORY22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY23 OFF;
ALTER TABLE HISTORY23 DROP CONSTRAINT HISTORY23CKC;
ALTER TABLE HISTORY23 ADD CONSTRAINT HISTORY23CKC
CHECK (H_W_ID BETWEEN 47301 AND 49450);
SET INTEGRITY FOR HISTORY23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY24 OFF;
ALTER TABLE HISTORY24 DROP CONSTRAINT HISTORY24CKC;
ALTER TABLE HISTORY24 ADD CONSTRAINT HISTORY24CKC
CHECK (H_W_ID BETWEEN 49451 AND 51600);
SET INTEGRITY FOR HISTORY24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY25 OFF;
ALTER TABLE HISTORY25 DROP CONSTRAINT HISTORY25CKC;
ALTER TABLE HISTORY25 ADD CONSTRAINT HISTORY25CKC
CHECK (H_W_ID BETWEEN 51601 AND 53750);
SET INTEGRITY FOR HISTORY25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY26 OFF;
ALTER TABLE HISTORY26 DROP CONSTRAINT HISTORY26CKC;
ALTER TABLE HISTORY26 ADD CONSTRAINT HISTORY26CKC
CHECK (H_W_ID BETWEEN 53751 AND 55900);
SET INTEGRITY FOR HISTORY26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY27 OFF;
ALTER TABLE HISTORY27 DROP CONSTRAINT HISTORY27CKC;
ALTER TABLE HISTORY27 ADD CONSTRAINT HISTORY27CKC
CHECK (H_W_ID BETWEEN 55901 AND 58050);
SET INTEGRITY FOR HISTORY27 ALL IMMEDIATE UNCHECKED;
connect reset;

connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY28 OFF;
ALTER TABLE HISTORY28 DROP CONSTRAINT HISTORY28CKC;
ALTER TABLE HISTORY28 ADD CONSTRAINT HISTORY28CKC
CHECK (H_W_ID BETWEEN 58051 AND 60200);
SET INTEGRITY FOR HISTORY28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY29 OFF;

```

```

ALTER TABLE HISTORY29 DROP CONSTRAINT HISTORY29CKC;
ALTER TABLE HISTORY29 ADD CONSTRAINT HISTORY29CKC
CHECK (H_W_ID BETWEEN 60201 AND 62350);
SET INTEGRITY FOR HISTORY29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY30 OFF;
ALTER TABLE HISTORY30 DROP CONSTRAINT HISTORY30CKC;
ALTER TABLE HISTORY30 ADD CONSTRAINT HISTORY30CKC
CHECK (H_W_ID BETWEEN 62351 AND 64500);
SET INTEGRITY FOR HISTORY30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY31 OFF;
ALTER TABLE HISTORY31 DROP CONSTRAINT HISTORY31CKC;
ALTER TABLE HISTORY31 ADD CONSTRAINT HISTORY31CKC
CHECK (H_W_ID BETWEEN 64501 AND 66650);
SET INTEGRITY FOR HISTORY31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY32 OFF;
ALTER TABLE HISTORY32 DROP CONSTRAINT HISTORY32CKC;
ALTER TABLE HISTORY32 ADD CONSTRAINT HISTORY32CKC
CHECK (H_W_ID BETWEEN 66651 AND 68800);
SET INTEGRITY FOR HISTORY32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY33 OFF;
ALTER TABLE HISTORY33 DROP CONSTRAINT HISTORY33CKC;
ALTER TABLE HISTORY33 ADD CONSTRAINT HISTORY33CKC
CHECK (H_W_ID BETWEEN 68801 AND 70950);
SET INTEGRITY FOR HISTORY33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY34 OFF;
ALTER TABLE HISTORY34 DROP CONSTRAINT HISTORY34CKC;
ALTER TABLE HISTORY34 ADD CONSTRAINT HISTORY34CKC
CHECK (H_W_ID BETWEEN 70951 AND 73100);
SET INTEGRITY FOR HISTORY34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY35 OFF;
ALTER TABLE HISTORY35 DROP CONSTRAINT HISTORY35CKC;
ALTER TABLE HISTORY35 ADD CONSTRAINT HISTORY35CKC
CHECK (H_W_ID BETWEEN 73101 AND 75250);
SET INTEGRITY FOR HISTORY35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY36 OFF;
ALTER TABLE HISTORY36 DROP CONSTRAINT HISTORY36CKC;
ALTER TABLE HISTORY36 ADD CONSTRAINT HISTORY36CKC
CHECK (H_W_ID BETWEEN 75251 AND 77400);
SET INTEGRITY FOR HISTORY36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY37 OFF;
ALTER TABLE HISTORY37 DROP CONSTRAINT HISTORY37CKC;
ALTER TABLE HISTORY37 ADD CONSTRAINT HISTORY37CKC
CHECK (H_W_ID BETWEEN 77401 AND 79550);
SET INTEGRITY FOR HISTORY37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR HISTORY38 OFF;
ALTER TABLE HISTORY38 DROP CONSTRAINT HISTORY38CKC;
ALTER TABLE HISTORY38 ADD CONSTRAINT HISTORY38CKC
CHECK (H_W_ID >= 79551);
SET INTEGRITY FOR HISTORY38 ALL IMMEDIATE UNCHECKED;
connect reset;

```

CRCONST NEW ORDER.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER1 OFF;
ALTER TABLE NEW_ORDER1 DROP CONSTRAINT
NEW_ORDER1CKC;
ALTER TABLE NEW_ORDER1 ADD CONSTRAINT NEW_ORDER1CKC
CHECK ((NO_W_ID BETWEEN 1 AND 2150) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER2 OFF;
ALTER TABLE NEW_ORDER2 DROP CONSTRAINT
NEW_ORDER2CKC;
ALTER TABLE NEW_ORDER2 ADD CONSTRAINT NEW_ORDER2CKC
CHECK ((NO_W_ID BETWEEN 2151 AND 4300) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER3 OFF;
ALTER TABLE NEW_ORDER3 DROP CONSTRAINT
NEW_ORDER3CKC;
ALTER TABLE NEW_ORDER3 ADD CONSTRAINT NEW_ORDER3CKC
CHECK ((NO_W_ID BETWEEN 4301 AND 6450) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER4 OFF;
ALTER TABLE NEW_ORDER4 DROP CONSTRAINT
NEW_ORDER4CKC;
ALTER TABLE NEW_ORDER4 ADD CONSTRAINT NEW_ORDER4CKC
CHECK ((NO_W_ID BETWEEN 6451 AND 8600) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER5 OFF;
ALTER TABLE NEW_ORDER5 DROP CONSTRAINT
NEW_ORDER5CKC;
ALTER TABLE NEW_ORDER5 ADD CONSTRAINT NEW_ORDER5CKC
CHECK ((NO_W_ID BETWEEN 8601 AND 10750) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER6 OFF;
ALTER TABLE NEW_ORDER6 DROP CONSTRAINT
NEW_ORDER6CKC;
ALTER TABLE NEW_ORDER6 ADD CONSTRAINT NEW_ORDER6CKC
CHECK ((NO_W_ID BETWEEN 10751 AND 12900) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER7 OFF;
ALTER TABLE NEW_ORDER7 DROP CONSTRAINT
NEW_ORDER7CKC;
ALTER TABLE NEW_ORDER7 ADD CONSTRAINT NEW_ORDER7CKC
CHECK ((NO_W_ID BETWEEN 12901 AND 15050) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER8 OFF;
ALTER TABLE NEW_ORDER8 DROP CONSTRAINT
NEW_ORDER8CKC;
ALTER TABLE NEW_ORDER8 ADD CONSTRAINT NEW_ORDER8CKC
CHECK ((NO_W_ID BETWEEN 15051 AND 17200) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER9 OFF;
ALTER TABLE NEW_ORDER9 DROP CONSTRAINT
NEW_ORDER9CKC;
ALTER TABLE NEW_ORDER9 ADD CONSTRAINT NEW_ORDER9CKC
CHECK ((NO_W_ID BETWEEN 17201 AND 19350) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER10 OFF;
ALTER TABLE NEW_ORDER10 DROP CONSTRAINT
NEW_ORDER10CKC;
ALTER TABLE NEW_ORDER10 ADD CONSTRAINT
NEW_ORDER10CKC CHECK ((NO_W_ID BETWEEN 19351 AND 21500)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER11 OFF;
ALTER TABLE NEW_ORDER11 DROP CONSTRAINT
NEW_ORDER11CKC;
ALTER TABLE NEW_ORDER11 ADD CONSTRAINT
NEW_ORDER11CKC CHECK ((NO_W_ID BETWEEN 21501 AND 23650)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER12 OFF;
ALTER TABLE NEW_ORDER12 DROP CONSTRAINT
NEW_ORDER12CKC;
ALTER TABLE NEW_ORDER12 ADD CONSTRAINT
NEW_ORDER12CKC CHECK ((NO_W_ID BETWEEN 23651 AND 25800)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER13 OFF;
ALTER TABLE NEW_ORDER13 DROP CONSTRAINT
NEW_ORDER13CKC;
ALTER TABLE NEW_ORDER13 ADD CONSTRAINT
NEW_ORDER13CKC CHECK ((NO_W_ID BETWEEN 25801 AND 27950)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER14 OFF;
ALTER TABLE NEW_ORDER14 DROP CONSTRAINT
NEW_ORDER14CKC;
ALTER TABLE NEW_ORDER14 ADD CONSTRAINT
NEW_ORDER14CKC CHECK ((NO_W_ID BETWEEN 27951 AND 30100)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER15 OFF;
ALTER TABLE NEW_ORDER15 DROP CONSTRAINT
NEW_ORDER15CKC;
ALTER TABLE NEW_ORDER15 ADD CONSTRAINT
NEW_ORDER15CKC CHECK ((NO_W_ID BETWEEN 30101 AND 32250)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER16 OFF;
ALTER TABLE NEW_ORDER16 DROP CONSTRAINT
NEW_ORDER16CKC;
ALTER TABLE NEW_ORDER16 ADD CONSTRAINT
NEW_ORDER16CKC CHECK ((NO_W_ID BETWEEN 32251 AND 34400)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER16 ALL IMMEDIATE UNCHECKED;
```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER17 OFF;
ALTER TABLE NEW_ORDER17 DROP CONSTRAINT
NEW_ORDER17CKC;
ALTER TABLE NEW_ORDER17 ADD CONSTRAINT
NEW_ORDER17CKC CHECK ((NO_W_ID BETWEEN 34401 AND 36550)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER18 OFF;
ALTER TABLE NEW_ORDER18 DROP CONSTRAINT
NEW_ORDER18CKC;
ALTER TABLE NEW_ORDER18 ADD CONSTRAINT
NEW_ORDER18CKC CHECK ((NO_W_ID BETWEEN 36551 AND 38700)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER19 OFF;
ALTER TABLE NEW_ORDER19 DROP CONSTRAINT
NEW_ORDER19CKC;
ALTER TABLE NEW_ORDER19 ADD CONSTRAINT
NEW_ORDER19CKC CHECK ((NO_W_ID BETWEEN 38701 AND 40850)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER20 OFF;
ALTER TABLE NEW_ORDER20 DROP CONSTRAINT
NEW_ORDER20CKC;
ALTER TABLE NEW_ORDER20 ADD CONSTRAINT
NEW_ORDER20CKC CHECK ((NO_W_ID BETWEEN 40851 AND 43000)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER21 OFF;
ALTER TABLE NEW_ORDER21 DROP CONSTRAINT
NEW_ORDER21CKC;
ALTER TABLE NEW_ORDER21 ADD CONSTRAINT
NEW_ORDER21CKC CHECK ((NO_W_ID BETWEEN 43001 AND 45150)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER22 OFF;
ALTER TABLE NEW_ORDER22 DROP CONSTRAINT
NEW_ORDER22CKC;
ALTER TABLE NEW_ORDER22 ADD CONSTRAINT
NEW_ORDER22CKC CHECK ((NO_W_ID BETWEEN 45151 AND 47300)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER23 OFF;
ALTER TABLE NEW_ORDER23 DROP CONSTRAINT
NEW_ORDER23CKC;
ALTER TABLE NEW_ORDER23 ADD CONSTRAINT
NEW_ORDER23CKC CHECK ((NO_W_ID BETWEEN 47301 AND 49450)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER24 OFF;
ALTER TABLE NEW_ORDER24 DROP CONSTRAINT
NEW_ORDER24CKC;
ALTER TABLE NEW_ORDER24 ADD CONSTRAINT
NEW_ORDER24CKC CHECK ((NO_W_ID BETWEEN 49451 AND 51600)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER24 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER25 OFF;
ALTER TABLE NEW_ORDER25 DROP CONSTRAINT
NEW_ORDER25CKC;
ALTER TABLE NEW_ORDER25 ADD CONSTRAINT
NEW_ORDER25CKC CHECK ((NO_W_ID BETWEEN 51601 AND 53750)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER26 OFF;
ALTER TABLE NEW_ORDER26 DROP CONSTRAINT
NEW_ORDER26CKC;
ALTER TABLE NEW_ORDER26 ADD CONSTRAINT
NEW_ORDER26CKC CHECK ((NO_W_ID BETWEEN 53751 AND 55900)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER27 OFF;
ALTER TABLE NEW_ORDER27 DROP CONSTRAINT
NEW_ORDER27CKC;
ALTER TABLE NEW_ORDER27 ADD CONSTRAINT
NEW_ORDER27CKC CHECK ((NO_W_ID BETWEEN 55901 AND 58050)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER28 OFF;
ALTER TABLE NEW_ORDER28 DROP CONSTRAINT
NEW_ORDER28CKC;
ALTER TABLE NEW_ORDER28 ADD CONSTRAINT
NEW_ORDER28CKC CHECK ((NO_W_ID BETWEEN 58051 AND 60200)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER29 OFF;
ALTER TABLE NEW_ORDER29 DROP CONSTRAINT
NEW_ORDER29CKC;
ALTER TABLE NEW_ORDER29 ADD CONSTRAINT
NEW_ORDER29CKC CHECK ((NO_W_ID BETWEEN 60201 AND 62350)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER30 OFF;
ALTER TABLE NEW_ORDER30 DROP CONSTRAINT
NEW_ORDER30CKC;
ALTER TABLE NEW_ORDER30 ADD CONSTRAINT
NEW_ORDER30CKC CHECK ((NO_W_ID BETWEEN 62351 AND 64500)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER31 OFF;
ALTER TABLE NEW_ORDER31 DROP CONSTRAINT
NEW_ORDER31CKC;
ALTER TABLE NEW_ORDER31 ADD CONSTRAINT
NEW_ORDER31CKC CHECK ((NO_W_ID BETWEEN 64501 AND 66650)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER32 OFF;
ALTER TABLE NEW_ORDER32 DROP CONSTRAINT
NEW_ORDER32CKC;
ALTER TABLE NEW_ORDER32 ADD CONSTRAINT
NEW_ORDER32CKC CHECK ((NO_W_ID BETWEEN 66651 AND 68800)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER32 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER33 OFF;
ALTER TABLE NEW_ORDER33 DROP CONSTRAINT
NEW_ORDER33CKC;
ALTER TABLE NEW_ORDER33 ADD CONSTRAINT
NEW_ORDER33CKC CHECK ((NO_W_ID BETWEEN 68801 AND 70950)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER34 OFF;
ALTER TABLE NEW_ORDER34 DROP CONSTRAINT
NEW_ORDER34CKC;
ALTER TABLE NEW_ORDER34 ADD CONSTRAINT
NEW_ORDER34CKC CHECK ((NO_W_ID BETWEEN 70951 AND 73100)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER35 OFF;
ALTER TABLE NEW_ORDER35 DROP CONSTRAINT
NEW_ORDER35CKC;
ALTER TABLE NEW_ORDER35 ADD CONSTRAINT
NEW_ORDER35CKC CHECK ((NO_W_ID BETWEEN 73101 AND 75250)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER35 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER36 OFF;
ALTER TABLE NEW_ORDER36 DROP CONSTRAINT
NEW_ORDER36CKC;
ALTER TABLE NEW_ORDER36 ADD CONSTRAINT
NEW_ORDER36CKC CHECK ((NO_W_ID BETWEEN 75251 AND 77400)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER37 OFF;
ALTER TABLE NEW_ORDER37 DROP CONSTRAINT
NEW_ORDER37CKC;
ALTER TABLE NEW_ORDER37 ADD CONSTRAINT
NEW_ORDER37CKC CHECK ((NO_W_ID BETWEEN 77401 AND 79550)
AND (NO_O_ID <= 3675));
SET INTEGRITY FOR NEW_ORDER37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER38 OFF;
ALTER TABLE NEW_ORDER38 DROP CONSTRAINT
NEW_ORDER38CKC;
ALTER TABLE NEW_ORDER38 ADD CONSTRAINT
NEW_ORDER38CKC CHECK ((NO_W_ID >= 79551) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR NEW_ORDER38 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER46 OFF;
ALTER TABLE NEW_ORDER46 DROP CONSTRAINT
NEW_ORDER46CKC;
ALTER TABLE NEW_ORDER46 ADD CONSTRAINT
NEW_ORDER46CKC CHECK ((NO_W_ID BETWEEN 1 AND 2150) AND
(NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER46 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER47 OFF;
ALTER TABLE NEW_ORDER47 DROP CONSTRAINT
NEW_ORDER47CKC;
ALTER TABLE NEW_ORDER47 ADD CONSTRAINT
NEW_ORDER47CKC CHECK ((NO_W_ID BETWEEN 2151 AND 4300)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER47 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER48 OFF;
ALTER TABLE NEW_ORDER48 DROP CONSTRAINT
NEW_ORDER48CKC;
ALTER TABLE NEW_ORDER48 ADD CONSTRAINT
NEW_ORDER48CKC CHECK ((NO_W_ID BETWEEN 4301 AND 6450)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER48 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER49 OFF;
ALTER TABLE NEW_ORDER49 DROP CONSTRAINT
NEW_ORDER49CKC;
ALTER TABLE NEW_ORDER49 ADD CONSTRAINT
NEW_ORDER49CKC CHECK ((NO_W_ID BETWEEN 6451 AND 8600)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER49 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER50 OFF;
ALTER TABLE NEW_ORDER50 DROP CONSTRAINT
NEW_ORDER50CKC;
ALTER TABLE NEW_ORDER50 ADD CONSTRAINT
NEW_ORDER50CKC CHECK ((NO_W_ID BETWEEN 8601 AND 10750)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER50 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER51 OFF;
ALTER TABLE NEW_ORDER51 DROP CONSTRAINT
NEW_ORDER51CKC;
ALTER TABLE NEW_ORDER51 ADD CONSTRAINT
NEW_ORDER51CKC CHECK ((NO_W_ID BETWEEN 10751 AND 12900)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER51 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER52 OFF;
ALTER TABLE NEW_ORDER52 DROP CONSTRAINT
NEW_ORDER52CKC;
ALTER TABLE NEW_ORDER52 ADD CONSTRAINT
NEW_ORDER52CKC CHECK ((NO_W_ID BETWEEN 12901 AND 15050)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER52 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER53 OFF;
ALTER TABLE NEW_ORDER53 DROP CONSTRAINT
NEW_ORDER53CKC;
ALTER TABLE NEW_ORDER53 ADD CONSTRAINT
NEW_ORDER53CKC CHECK ((NO_W_ID BETWEEN 15051 AND 17200)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER53 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER54 OFF;
ALTER TABLE NEW_ORDER54 DROP CONSTRAINT
NEW_ORDER54CKC;
ALTER TABLE NEW_ORDER54 ADD CONSTRAINT
NEW_ORDER54CKC CHECK ((NO_W_ID BETWEEN 17201 AND 19350)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER54 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER55 OFF;
ALTER TABLE NEW_ORDER55 DROP CONSTRAINT
NEW_ORDER55CKC;
ALTER TABLE NEW_ORDER55 ADD CONSTRAINT
NEW_ORDER55CKC CHECK ((NO_W_ID BETWEEN 19351 AND 21500)
AND (NO_O_ID >= 3676));

```

```

SET INTEGRITY FOR NEW_ORDER55 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER56 OFF;
ALTER TABLE NEW_ORDER56 DROP CONSTRAINT
NEW_ORDER56CKC;
ALTER TABLE NEW_ORDER56 ADD CONSTRAINT
NEW_ORDER56CKC CHECK ((NO_W_ID BETWEEN 21501 AND 23650)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER56 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER57 OFF;
ALTER TABLE NEW_ORDER57 DROP CONSTRAINT
NEW_ORDER57CKC;
ALTER TABLE NEW_ORDER57 ADD CONSTRAINT
NEW_ORDER57CKC CHECK ((NO_W_ID BETWEEN 23651 AND 25800)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER57 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER58 OFF;
ALTER TABLE NEW_ORDER58 DROP CONSTRAINT
NEW_ORDER58CKC;
ALTER TABLE NEW_ORDER58 ADD CONSTRAINT
NEW_ORDER58CKC CHECK ((NO_W_ID BETWEEN 25801 AND 27950)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER58 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER59 OFF;
ALTER TABLE NEW_ORDER59 DROP CONSTRAINT
NEW_ORDER59CKC;
ALTER TABLE NEW_ORDER59 ADD CONSTRAINT
NEW_ORDER59CKC CHECK ((NO_W_ID BETWEEN 27951 AND 30100)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER59 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER60 OFF;
ALTER TABLE NEW_ORDER60 DROP CONSTRAINT
NEW_ORDER60CKC;
ALTER TABLE NEW_ORDER60 ADD CONSTRAINT
NEW_ORDER60CKC CHECK ((NO_W_ID BETWEEN 30101 AND 32250)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER60 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER61 OFF;
ALTER TABLE NEW_ORDER61 DROP CONSTRAINT
NEW_ORDER61CKC;
ALTER TABLE NEW_ORDER61 ADD CONSTRAINT
NEW_ORDER61CKC CHECK ((NO_W_ID BETWEEN 32251 AND 34400)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER61 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER62 OFF;
ALTER TABLE NEW_ORDER62 DROP CONSTRAINT
NEW_ORDER62CKC;
ALTER TABLE NEW_ORDER62 ADD CONSTRAINT
NEW_ORDER62CKC CHECK ((NO_W_ID BETWEEN 34401 AND 36550)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER62 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER63 OFF;
ALTER TABLE NEW_ORDER63 DROP CONSTRAINT
NEW_ORDER63CKC;
ALTER TABLE NEW_ORDER63 ADD CONSTRAINT
NEW_ORDER63CKC CHECK ((NO_W_ID BETWEEN 36551 AND 38700)
AND (NO_O_ID >= 3676));

```

```

SET INTEGRITY FOR NEW_ORDER63 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER64 OFF;
ALTER TABLE NEW_ORDER64 DROP CONSTRAINT
NEW_ORDER64CKC;
ALTER TABLE NEW_ORDER64 ADD CONSTRAINT
NEW_ORDER64CKC CHECK ((NO_W_ID BETWEEN 38701 AND 40850)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER64 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER65 OFF;
ALTER TABLE NEW_ORDER65 DROP CONSTRAINT
NEW_ORDER65CKC;
ALTER TABLE NEW_ORDER65 ADD CONSTRAINT
NEW_ORDER65CKC CHECK ((NO_W_ID BETWEEN 40851 AND 43000)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER65 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER66 OFF;
ALTER TABLE NEW_ORDER66 DROP CONSTRAINT
NEW_ORDER66CKC;
ALTER TABLE NEW_ORDER66 ADD CONSTRAINT
NEW_ORDER66CKC CHECK ((NO_W_ID BETWEEN 43001 AND 45150)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER66 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER67 OFF;
ALTER TABLE NEW_ORDER67 DROP CONSTRAINT
NEW_ORDER67CKC;
ALTER TABLE NEW_ORDER67 ADD CONSTRAINT
NEW_ORDER67CKC CHECK ((NO_W_ID BETWEEN 45151 AND 47300)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER67 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER68 OFF;
ALTER TABLE NEW_ORDER68 DROP CONSTRAINT
NEW_ORDER68CKC;
ALTER TABLE NEW_ORDER68 ADD CONSTRAINT
NEW_ORDER68CKC CHECK ((NO_W_ID BETWEEN 47301 AND 49450)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER68 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER69 OFF;
ALTER TABLE NEW_ORDER69 DROP CONSTRAINT
NEW_ORDER69CKC;
ALTER TABLE NEW_ORDER69 ADD CONSTRAINT
NEW_ORDER69CKC CHECK ((NO_W_ID BETWEEN 49451 AND 51600)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER69 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER70 OFF;
ALTER TABLE NEW_ORDER70 DROP CONSTRAINT
NEW_ORDER70CKC;
ALTER TABLE NEW_ORDER70 ADD CONSTRAINT
NEW_ORDER70CKC CHECK ((NO_W_ID BETWEEN 51601 AND 53750)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER70 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER71 OFF;
ALTER TABLE NEW_ORDER71 DROP CONSTRAINT
NEW_ORDER71CKC;
ALTER TABLE NEW_ORDER71 ADD CONSTRAINT
NEW_ORDER71CKC CHECK ((NO_W_ID BETWEEN 53751 AND 55900)
AND (NO_O_ID >= 3676));

```

```

SET INTEGRITY FOR NEW_ORDER71 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER72 OFF;
ALTER TABLE NEW_ORDER72 DROP CONSTRAINT
NEW_ORDER72CKC;
ALTER TABLE NEW_ORDER72 ADD CONSTRAINT
NEW_ORDER72CKC CHECK ((NO_W_ID BETWEEN 55901 AND 58050)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER72 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER73 OFF;
ALTER TABLE NEW_ORDER73 DROP CONSTRAINT
NEW_ORDER73CKC;
ALTER TABLE NEW_ORDER73 ADD CONSTRAINT
NEW_ORDER73CKC CHECK ((NO_W_ID BETWEEN 58051 AND 60200)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER73 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER74 OFF;
ALTER TABLE NEW_ORDER74 DROP CONSTRAINT
NEW_ORDER74CKC;
ALTER TABLE NEW_ORDER74 ADD CONSTRAINT
NEW_ORDER74CKC CHECK ((NO_W_ID BETWEEN 60201 AND 62350)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER74 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER75 OFF;
ALTER TABLE NEW_ORDER75 DROP CONSTRAINT
NEW_ORDER75CKC;
ALTER TABLE NEW_ORDER75 ADD CONSTRAINT
NEW_ORDER75CKC CHECK ((NO_W_ID BETWEEN 62351 AND 64500)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER75 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER76 OFF;
ALTER TABLE NEW_ORDER76 DROP CONSTRAINT
NEW_ORDER76CKC;
ALTER TABLE NEW_ORDER76 ADD CONSTRAINT
NEW_ORDER76CKC CHECK ((NO_W_ID BETWEEN 64501 AND 66650)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER76 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER77 OFF;
ALTER TABLE NEW_ORDER77 DROP CONSTRAINT
NEW_ORDER77CKC;
ALTER TABLE NEW_ORDER77 ADD CONSTRAINT
NEW_ORDER77CKC CHECK ((NO_W_ID BETWEEN 66651 AND 68800)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER77 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER78 OFF;
ALTER TABLE NEW_ORDER78 DROP CONSTRAINT
NEW_ORDER78CKC;
ALTER TABLE NEW_ORDER78 ADD CONSTRAINT
NEW_ORDER78CKC CHECK ((NO_W_ID BETWEEN 68801 AND 70950)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER78 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER79 OFF;
ALTER TABLE NEW_ORDER79 DROP CONSTRAINT
NEW_ORDER79CKC;
ALTER TABLE NEW_ORDER79 ADD CONSTRAINT
NEW_ORDER79CKC CHECK ((NO_W_ID BETWEEN 70951 AND 73100)
AND (NO_O_ID >= 3676));

```

```

SET INTEGRITY FOR NEW_ORDER79 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER80 OFF;
ALTER TABLE NEW_ORDER80 DROP CONSTRAINT
NEW_ORDER80CKC;
ALTER TABLE NEW_ORDER80 ADD CONSTRAINT
NEW_ORDER80CKC CHECK ((NO_W_ID BETWEEN 73101 AND 75250)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER80 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER81 OFF;
ALTER TABLE NEW_ORDER81 DROP CONSTRAINT
NEW_ORDER81CKC;
ALTER TABLE NEW_ORDER81 ADD CONSTRAINT
NEW_ORDER81CKC CHECK ((NO_W_ID BETWEEN 75251 AND 77400)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER81 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER82 OFF;
ALTER TABLE NEW_ORDER82 DROP CONSTRAINT
NEW_ORDER82CKC;
ALTER TABLE NEW_ORDER82 ADD CONSTRAINT
NEW_ORDER82CKC CHECK ((NO_W_ID BETWEEN 77401 AND 79550)
AND (NO_O_ID >= 3676));
SET INTEGRITY FOR NEW_ORDER82 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR NEW_ORDER83 OFF;
ALTER TABLE NEW_ORDER83 DROP CONSTRAINT
NEW_ORDER83CKC;
ALTER TABLE NEW_ORDER83 ADD CONSTRAINT
NEW_ORDER83CKC CHECK ((NO_W_ID >= 79551) AND (NO_O_ID >=
3676));
SET INTEGRITY FOR NEW_ORDER83 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 2150);
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 2151 AND 4300);
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 4301 AND 6450);
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 6451 AND 8600);
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 8601 AND 10750);
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 10751 AND 12900);
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 12901 AND 15050);
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 15051 AND 17200);
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 17201 AND 19350);
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 19351 AND 21500);
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 21501 AND 23650);
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 23651 AND 25800);
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 25801 AND 27950);
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 27951 AND 30100);
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 30101 AND 32250);
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 32251 AND 34400);
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 34401 AND 36550);
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 36551 AND 38700);
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 38701 AND 40850);
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 40851 AND 43000);
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 43001 AND 45150);
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 45151 AND 47300);
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 47301 AND 49450);
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 49451 AND 51600);
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 51601 AND 53750);

```



```

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 53751 AND 55900);
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 55901 AND 58050);
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 58051 AND 60200);
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 60201 AND 62350);
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 62351 AND 64500);
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 64501 AND 66650);
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 66651 AND 68800);
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 68801 AND 70950);
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 70951 AND 73100);
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 73101 AND 75250);
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 75251 AND 77400);
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 77401 AND 79550);
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 >= 79551);
SET INTEGRITY FOR ORDERS38 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 2150);
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 2151 AND 4300);
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 4301 AND 6450);
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 6451 AND 8600);
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 8601 AND 10750);
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 10751 AND 12900);

```

```

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 12901 AND 15050);
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 15051 AND 17200);
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 17201 AND 19350);
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 19351 AND 21500);
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 21501 AND 23650);
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 23651 AND 25800);
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 25801 AND 27950);
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 27951 AND 30100);
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 30101 AND 32250);
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 32251 AND 34400);
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 34401 AND 36550);
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 36551 AND 38700);
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 38701 AND 40850);
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 40851 AND 43000);
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 43001 AND 45150);
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 45151 AND 47300);
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 47301 AND 49450);
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 49451 AND 51600);
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 51601 AND 53750);
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 53751 AND 55900);
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 55901 AND 58050);
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 58051 AND 60200);
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 60201 AND 62350);
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 62351 AND 64500);
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 64501 AND 66650);
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 66651 AND 68800);
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 68801 AND 70950);
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 70951 AND 73100);
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 73101 AND 75250);
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 75251 AND 77400);
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 77401 AND 79550);
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 >= 79551);
SET INTEGRITY FOR ORDER_LINE38 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 1075);
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 1076 AND 2150);
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 2151 AND 3225);
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 3226 AND 4300);
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 4301 AND 5375);
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 5376 AND 6450);
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 6451 AND 7525);
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 7526 AND 8600);
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 8601 AND 9675);
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 9676 AND 10750);
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 10751 AND 11825);
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 11826 AND 12900);
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 12901 AND 13975);
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 13976 AND 15050);

```

```

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 15051 AND 16125);
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 16126 AND 17200);
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 17201 AND 18275);
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 18276 AND 19350);
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 19351 AND 20425);
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 20426 AND 21500);
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 21501 AND 22575);
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 22576 AND 23650);
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 23651 AND 24725);
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 24726 AND 25800);
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 25801 AND 26875);
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 26876 AND 27950);
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 27951 AND 29025);
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 29026 AND 30100);
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 30101 AND 31175);
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 31176 AND 32250);
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 32251 AND 33325);
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 33326 AND 34400);
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 34401 AND 35475);
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 35476 AND 36550);
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 36551 AND 37625);
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 37626 AND 38700);
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 38701 AND 39775);
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 39776 AND 40850);
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 40851 AND 41925);
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 41926 AND 43000);
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 43001 AND 44075);
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 44076 AND 45150);
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 45151 AND 46225);
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 46226 AND 47300);
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 47301 AND 48375);
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 48376 AND 49450);
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 49451 AND 50525);
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 50526 AND 51600);
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 51601 AND 52675);
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 52676 AND 53750);
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 53751 AND 54825);
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 54826 AND 55900);
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 55901 AND 56975);
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 56976 AND 58050);
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 58051 AND 59125);

```

```

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 59126 AND 60200);
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 60201 AND 61275);
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 61276 AND 62350);
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 62351 AND 63425);
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 63426 AND 64500);
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 64501 AND 65575);
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 65576 AND 66650);
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 66651 AND 67725);
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 67726 AND 68800);
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 68801 AND 69875);
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 69876 AND 70950);
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 70951 AND 72025);
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 72026 AND 73100);
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 73101 AND 74175);
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 74176 AND 75250);
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 75251 AND 76325);
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 76326 AND 77400);
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 77401 AND 78475);
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 78476 AND 79550);
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 79551 AND 80625);
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 >= 80626);
SET INTEGRITY FOR STOCK76 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 2150);
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 2151 AND 4300);
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 4301 AND 6450);
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 6451 AND 8600);
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 8601 AND 10750);
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 10751 AND 12900);
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 12901 AND 15050);
SET INTEGRITY FOR WAREHOUSE7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE8 OFF;
ALTER TABLE WAREHOUSE8 DROP CONSTRAINT
WAREHOUSE8CKC;
ALTER TABLE WAREHOUSE8 ADD CONSTRAINT WAREHOUSE8CKC
CHECK (W_ID BETWEEN 15051 AND 17200);
SET INTEGRITY FOR WAREHOUSE8 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE9 OFF;
ALTER TABLE WAREHOUSE9 DROP CONSTRAINT
WAREHOUSE9CKC;
ALTER TABLE WAREHOUSE9 ADD CONSTRAINT WAREHOUSE9CKC
CHECK (W_ID BETWEEN 17201 AND 19350);
SET INTEGRITY FOR WAREHOUSE9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE10 OFF;
ALTER TABLE WAREHOUSE10 DROP CONSTRAINT
WAREHOUSE10CKC;
ALTER TABLE WAREHOUSE10 ADD CONSTRAINT
WAREHOUSE10CKC CHECK (W_ID BETWEEN 19351 AND 21500);
SET INTEGRITY FOR WAREHOUSE10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE11 OFF;
ALTER TABLE WAREHOUSE11 DROP CONSTRAINT
WAREHOUSE11CKC;
ALTER TABLE WAREHOUSE11 ADD CONSTRAINT
WAREHOUSE11CKC CHECK (W_ID BETWEEN 21501 AND 23650);
SET INTEGRITY FOR WAREHOUSE11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE12 OFF;
ALTER TABLE WAREHOUSE12 DROP CONSTRAINT
WAREHOUSE12CKC;
ALTER TABLE WAREHOUSE12 ADD CONSTRAINT
WAREHOUSE12CKC CHECK (W_ID BETWEEN 23651 AND 25800);
SET INTEGRITY FOR WAREHOUSE12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE13 OFF;
ALTER TABLE WAREHOUSE13 DROP CONSTRAINT
WAREHOUSE13CKC;
ALTER TABLE WAREHOUSE13 ADD CONSTRAINT
WAREHOUSE13CKC CHECK (W_ID BETWEEN 25801 AND 27950);
SET INTEGRITY FOR WAREHOUSE13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE14 OFF;
ALTER TABLE WAREHOUSE14 DROP CONSTRAINT
WAREHOUSE14CKC;
ALTER TABLE WAREHOUSE14 ADD CONSTRAINT
WAREHOUSE14CKC CHECK (W_ID BETWEEN 27951 AND 30100);
SET INTEGRITY FOR WAREHOUSE14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE15 OFF;
ALTER TABLE WAREHOUSE15 DROP CONSTRAINT
WAREHOUSE15CKC;
ALTER TABLE WAREHOUSE15 ADD CONSTRAINT
WAREHOUSE15CKC CHECK (W_ID BETWEEN 30101 AND 32250);
SET INTEGRITY FOR WAREHOUSE15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE16 OFF;
ALTER TABLE WAREHOUSE16 DROP CONSTRAINT
WAREHOUSE16CKC;
ALTER TABLE WAREHOUSE16 ADD CONSTRAINT
WAREHOUSE16CKC CHECK (W_ID BETWEEN 32251 AND 34400);
SET INTEGRITY FOR WAREHOUSE16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE17 OFF;
ALTER TABLE WAREHOUSE17 DROP CONSTRAINT
WAREHOUSE17CKC;
ALTER TABLE WAREHOUSE17 ADD CONSTRAINT
WAREHOUSE17CKC CHECK (W_ID BETWEEN 34401 AND 36550);
SET INTEGRITY FOR WAREHOUSE17 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE18 OFF;
ALTER TABLE WAREHOUSE18 DROP CONSTRAINT
WAREHOUSE18CKC;
ALTER TABLE WAREHOUSE18 ADD CONSTRAINT
WAREHOUSE18CKC CHECK (W_ID BETWEEN 36551 AND 38700);
SET INTEGRITY FOR WAREHOUSE18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE19 OFF;
ALTER TABLE WAREHOUSE19 DROP CONSTRAINT
WAREHOUSE19CKC;
ALTER TABLE WAREHOUSE19 ADD CONSTRAINT
WAREHOUSE19CKC CHECK (W_ID BETWEEN 38701 AND 40850);
SET INTEGRITY FOR WAREHOUSE19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE20 OFF;
ALTER TABLE WAREHOUSE20 DROP CONSTRAINT
WAREHOUSE20CKC;
ALTER TABLE WAREHOUSE20 ADD CONSTRAINT
WAREHOUSE20CKC CHECK (W_ID BETWEEN 40851 AND 43000);
SET INTEGRITY FOR WAREHOUSE20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE21 OFF;
ALTER TABLE WAREHOUSE21 DROP CONSTRAINT
WAREHOUSE21CKC;
ALTER TABLE WAREHOUSE21 ADD CONSTRAINT
WAREHOUSE21CKC CHECK (W_ID BETWEEN 43001 AND 45150);
SET INTEGRITY FOR WAREHOUSE21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE22 OFF;
ALTER TABLE WAREHOUSE22 DROP CONSTRAINT
WAREHOUSE22CKC;
ALTER TABLE WAREHOUSE22 ADD CONSTRAINT
WAREHOUSE22CKC CHECK (W_ID BETWEEN 45151 AND 47300);
SET INTEGRITY FOR WAREHOUSE22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE23 OFF;
ALTER TABLE WAREHOUSE23 DROP CONSTRAINT
WAREHOUSE23CKC;
ALTER TABLE WAREHOUSE23 ADD CONSTRAINT
WAREHOUSE23CKC CHECK (W_ID BETWEEN 47301 AND 49450);
SET INTEGRITY FOR WAREHOUSE23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE24 OFF;
ALTER TABLE WAREHOUSE24 DROP CONSTRAINT
WAREHOUSE24CKC;
ALTER TABLE WAREHOUSE24 ADD CONSTRAINT
WAREHOUSE24CKC CHECK (W_ID BETWEEN 49451 AND 51600);
SET INTEGRITY FOR WAREHOUSE24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE25 OFF;
ALTER TABLE WAREHOUSE25 DROP CONSTRAINT
WAREHOUSE25CKC;
ALTER TABLE WAREHOUSE25 ADD CONSTRAINT
WAREHOUSE25CKC CHECK (W_ID BETWEEN 51601 AND 53750);
SET INTEGRITY FOR WAREHOUSE25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE26 OFF;
ALTER TABLE WAREHOUSE26 DROP CONSTRAINT
WAREHOUSE26CKC;
ALTER TABLE WAREHOUSE26 ADD CONSTRAINT
WAREHOUSE26CKC CHECK (W_ID BETWEEN 53751 AND 55900);
SET INTEGRITY FOR WAREHOUSE26 ALL IMMEDIATE UNCHECKED;

```



```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE27 OFF;
ALTER TABLE WAREHOUSE27 DROP CONSTRAINT
WAREHOUSE27CKC;
ALTER TABLE WAREHOUSE27 ADD CONSTRAINT
WAREHOUSE27CKC CHECK (W_ID BETWEEN 55901 AND 58050);
SET INTEGRITY FOR WAREHOUSE27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE28 OFF;
ALTER TABLE WAREHOUSE28 DROP CONSTRAINT
WAREHOUSE28CKC;
ALTER TABLE WAREHOUSE28 ADD CONSTRAINT
WAREHOUSE28CKC CHECK (W_ID BETWEEN 58051 AND 60200);
SET INTEGRITY FOR WAREHOUSE28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE29 OFF;
ALTER TABLE WAREHOUSE29 DROP CONSTRAINT
WAREHOUSE29CKC;
ALTER TABLE WAREHOUSE29 ADD CONSTRAINT
WAREHOUSE29CKC CHECK (W_ID BETWEEN 60201 AND 62350);
SET INTEGRITY FOR WAREHOUSE29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE30 OFF;
ALTER TABLE WAREHOUSE30 DROP CONSTRAINT
WAREHOUSE30CKC;
ALTER TABLE WAREHOUSE30 ADD CONSTRAINT
WAREHOUSE30CKC CHECK (W_ID BETWEEN 62351 AND 64500);
SET INTEGRITY FOR WAREHOUSE30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE31 OFF;
ALTER TABLE WAREHOUSE31 DROP CONSTRAINT
WAREHOUSE31CKC;
ALTER TABLE WAREHOUSE31 ADD CONSTRAINT
WAREHOUSE31CKC CHECK (W_ID BETWEEN 64501 AND 66650);
SET INTEGRITY FOR WAREHOUSE31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE32 OFF;
ALTER TABLE WAREHOUSE32 DROP CONSTRAINT
WAREHOUSE32CKC;
ALTER TABLE WAREHOUSE32 ADD CONSTRAINT
WAREHOUSE32CKC CHECK (W_ID BETWEEN 66651 AND 68800);
SET INTEGRITY FOR WAREHOUSE32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE33 OFF;
ALTER TABLE WAREHOUSE33 DROP CONSTRAINT
WAREHOUSE33CKC;
ALTER TABLE WAREHOUSE33 ADD CONSTRAINT
WAREHOUSE33CKC CHECK (W_ID BETWEEN 68801 AND 70950);
SET INTEGRITY FOR WAREHOUSE33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE34 OFF;
ALTER TABLE WAREHOUSE34 DROP CONSTRAINT
WAREHOUSE34CKC;
ALTER TABLE WAREHOUSE34 ADD CONSTRAINT
WAREHOUSE34CKC CHECK (W_ID BETWEEN 70951 AND 73100);
SET INTEGRITY FOR WAREHOUSE34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE35 OFF;
ALTER TABLE WAREHOUSE35 DROP CONSTRAINT
WAREHOUSE35CKC;
ALTER TABLE WAREHOUSE35 ADD CONSTRAINT
WAREHOUSE35CKC CHECK (W_ID BETWEEN 73101 AND 75250);
SET INTEGRITY FOR WAREHOUSE35 ALL IMMEDIATE UNCHECKED;

```

```

connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE36 OFF;
ALTER TABLE WAREHOUSE36 DROP CONSTRAINT
WAREHOUSE36CKC;
ALTER TABLE WAREHOUSE36 ADD CONSTRAINT
WAREHOUSE36CKC CHECK (W_ID BETWEEN 75251 AND 77400);
SET INTEGRITY FOR WAREHOUSE36 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE37 OFF;
ALTER TABLE WAREHOUSE37 DROP CONSTRAINT
WAREHOUSE37CKC;
ALTER TABLE WAREHOUSE37 ADD CONSTRAINT
WAREHOUSE37CKC CHECK (W_ID BETWEEN 77401 AND 79550);
SET INTEGRITY FOR WAREHOUSE37 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR WAREHOUSE38 OFF;
ALTER TABLE WAREHOUSE38 DROP CONSTRAINT
WAREHOUSE38CKC;
ALTER TABLE WAREHOUSE38 ADD CONSTRAINT
WAREHOUSE38CKC CHECK (W_ID >= 79551);
SET INTEGRITY FOR WAREHOUSE38 ALL IMMEDIATE UNCHECKED;
connect reset;

```

CRIDX_CUST_IDXB.ddl

```

connect to TPCC in share mode;
DROP INDEX CUST_IDXB1;
CREATE INDEX CUST_IDXB1
ON CUSTOMER1(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB2;
CREATE INDEX CUST_IDXB2
ON CUSTOMER2(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB3;
CREATE INDEX CUST_IDXB3
ON CUSTOMER3(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB4;
CREATE INDEX CUST_IDXB4
ON CUSTOMER4(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB5;
CREATE INDEX CUST_IDXB5
ON CUSTOMER5(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB6;
CREATE INDEX CUST_IDXB6
ON CUSTOMER6(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB7;
CREATE INDEX CUST_IDXB7
ON CUSTOMER7(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;

```

```

connect to TPCC in share mode;
DROP INDEX CUST_IDXB8;
CREATE INDEX CUST_IDXB8
    ON CUSTOMER8(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB9;
CREATE INDEX CUST_IDXB9
    ON CUSTOMER9(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB10;
CREATE INDEX CUST_IDXB10
    ON CUSTOMER10(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB11;
CREATE INDEX CUST_IDXB11
    ON CUSTOMER11(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB12;
CREATE INDEX CUST_IDXB12
    ON CUSTOMER12(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB13;
CREATE INDEX CUST_IDXB13
    ON CUSTOMER13(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB14;
CREATE INDEX CUST_IDXB14
    ON CUSTOMER14(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB15;
CREATE INDEX CUST_IDXB15
    ON CUSTOMER15(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB16;
CREATE INDEX CUST_IDXB16
    ON CUSTOMER16(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB17;
CREATE INDEX CUST_IDXB17
    ON CUSTOMER17(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB18;
CREATE INDEX CUST_IDXB18
    ON CUSTOMER18(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB19;
CREATE INDEX CUST_IDXB19
    ON CUSTOMER19(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;

```

```

connect to TPCC in share mode;
DROP INDEX CUST_IDXB20;
CREATE INDEX CUST_IDXB20
    ON CUSTOMER20(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB21;
CREATE INDEX CUST_IDXB21
    ON CUSTOMER21(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB22;
CREATE INDEX CUST_IDXB22
    ON CUSTOMER22(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB23;
CREATE INDEX CUST_IDXB23
    ON CUSTOMER23(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB24;
CREATE INDEX CUST_IDXB24
    ON CUSTOMER24(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB25;
CREATE INDEX CUST_IDXB25
    ON CUSTOMER25(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB26;
CREATE INDEX CUST_IDXB26
    ON CUSTOMER26(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB27;
CREATE INDEX CUST_IDXB27
    ON CUSTOMER27(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;

connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB28;
CREATE INDEX CUST_IDXB28
    ON CUSTOMER28(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB29;
CREATE INDEX CUST_IDXB29
    ON CUSTOMER29(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB30;
CREATE INDEX CUST_IDXB30
    ON CUSTOMER30(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB31;
CREATE INDEX CUST_IDXB31
    ON CUSTOMER31(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;

```

```

connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB32;
CREATE INDEX CUST_IDXB32
    ON CUSTOMER32(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB33;
CREATE INDEX CUST_IDXB33
    ON CUSTOMER33(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB34;
CREATE INDEX CUST_IDXB34
    ON CUSTOMER34(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB35;
CREATE INDEX CUST_IDXB35
    ON CUSTOMER35(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;

connect to TPCC in share mode;
DROP INDEX CUST_IDXB36;
CREATE INDEX CUST_IDXB36
    ON CUSTOMER36(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB37;
CREATE INDEX CUST_IDXB37
    ON CUSTOMER37(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB38;
CREATE INDEX CUST_IDXB38
    ON CUSTOMER38(C_LAST, C_W_ID, C_D_ID, C_FIRST,
C_ID) PCTFREE 0;
connect reset;

```

CRIDX ORDR_IDXB.ddl

```

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

```

```

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB5;
CREATE INDEX ORDR_IDXB5
    ON ORDERS5(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB6;
CREATE INDEX ORDR_IDXB6
    ON ORDERS6(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB7;
CREATE INDEX ORDR_IDXB7
    ON ORDERS7(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB8;
CREATE INDEX ORDR_IDXB8
    ON ORDERS8(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB9;
CREATE INDEX ORDR_IDXB9
    ON ORDERS9(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB10;
CREATE INDEX ORDR_IDXB10
    ON ORDERS10(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB11;
CREATE INDEX ORDR_IDXB11
    ON ORDERS11(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB12;
CREATE INDEX ORDR_IDXB12
    ON ORDERS12(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB13;
CREATE INDEX ORDR_IDXB13
    ON ORDERS13(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB14;
CREATE INDEX ORDR_IDXB14
    ON ORDERS14(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB15;
CREATE INDEX ORDR_IDXB15
    ON ORDERS15(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB16;
CREATE INDEX ORDR_IDXB16
    ON ORDERS16(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

```

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB17;
CREATE INDEX ORDR_IDXB17
    ON ORDERS17(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB18;
CREATE INDEX ORDR_IDXB18
    ON ORDERS18(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB19;
CREATE INDEX ORDR_IDXB19
    ON ORDERS19(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB20;
CREATE INDEX ORDR_IDXB20
    ON ORDERS20(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB21;
CREATE INDEX ORDR_IDXB21
    ON ORDERS21(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB22;
CREATE INDEX ORDR_IDXB22
    ON ORDERS22(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB23;
CREATE INDEX ORDR_IDXB23
    ON ORDERS23(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB24;
CREATE INDEX ORDR_IDXB24
    ON ORDERS24(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB25;
CREATE INDEX ORDR_IDXB25
    ON ORDERS25(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB26;
CREATE INDEX ORDR_IDXB26
    ON ORDERS26(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB27;
CREATE INDEX ORDR_IDXB27
    ON ORDERS27(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB28;
CREATE INDEX ORDR_IDXB28
    ON ORDERS28(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

```

connect to TPCC in share mode;
DROP INDEX ORDR_IDXB29;
CREATE INDEX ORDR_IDXB29
    ON ORDERS29(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB30;
CREATE INDEX ORDR_IDXB30
    ON ORDERS30(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB31;
CREATE INDEX ORDR_IDXB31
    ON ORDERS31(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB32;
CREATE INDEX ORDR_IDXB32
    ON ORDERS32(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB33;
CREATE INDEX ORDR_IDXB33
    ON ORDERS33(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB34;
CREATE INDEX ORDR_IDXB34
    ON ORDERS34(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB35;
CREATE INDEX ORDR_IDXB35
    ON ORDERS35(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB36;
CREATE INDEX ORDR_IDXB36
    ON ORDERS36(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB37;
CREATE INDEX ORDR_IDXB37
    ON ORDERS37(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB38;
CREATE INDEX ORDR_IDXB38
    ON ORDERS38(O_C_ID, O_W_ID, O_D_ID, O_ID DESC)
PCTFREE 20 LEVEL2 PCTFREE 20;
connect reset;

```

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_01
INDEX IN is_customer_01
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 1 ENDING AT 2150,
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_02
INDEX IN is_customer_02
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 2151 ENDING AT 4300,
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_03
INDEX IN is_customer_03
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 4301 ENDING AT 6450,
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_04
INDEX IN is_customer_04
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 6451 ENDING AT 8600,
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_05
INDEX IN is_customer_05
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 8601 ENDING AT 10750,
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_06
INDEX IN is_customer_06
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 10751 ENDING AT 12900,
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_07
INDEX IN is_customer_07
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 12901 ENDING AT 15050,
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_08
INDEX IN is_customer_08
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 15051 ENDING AT 17200,
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_09
INDEX IN is_customer_09
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 17201 ENDING AT 19350,
  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_10
INDEX IN is_customer_10
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 19351 ENDING AT 21500,
  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_11
INDEX IN is_customer_11
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 21501 ENDING AT 23650,
  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_12
INDEX IN is_customer_12
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 23651 ENDING AT 25800,
  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_13
INDEX IN is_customer_13
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 25801 ENDING AT 27950,
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_14
INDEX IN is_customer_14
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 27951 ENDING AT 30100,
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_15
INDEX IN is_customer_15
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 30101 ENDING AT 32250,
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_16
INDEX IN is_customer_16
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 32251 ENDING AT 34400,
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_17
INDEX IN is_customer_17
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 34401 ENDING AT 36550,
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_18
INDEX IN is_customer_18
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 36551 ENDING AT 38700,
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_19
INDEX IN is_customer_19
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 38701 ENDING AT 40850,
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_20
INDEX IN is_customer_20
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 40851 ENDING AT 43000,
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_21
INDEX IN is_customer_21
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 43001 ENDING AT 45150,
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_22
INDEX IN is_customer_22
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 45151 ENDING AT 47300,
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_23
INDEX IN is_customer_23
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 47301 ENDING AT 49450,
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_24
INDEX IN is_customer_24
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 49451 ENDING AT 51600,
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_25
INDEX IN is_customer_25
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 51601 ENDING AT 53750,
  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_26
INDEX IN is_customer_26
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 53751 ENDING AT 55900,
  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_27
INDEX IN is_customer_27

```

```

ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 55901 ENDING AT 58050,
  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_28
INDEX IN is_customer_28
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 58051 ENDING AT 60200,
  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_29
INDEX IN is_customer_29
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,

```

```

C_W_ID STARTING FROM 60201 ENDING AT 62350,
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_30
INDEX IN is_customer_30
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 62351 ENDING AT 64500,
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_31
INDEX IN is_customer_31
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 64501 ENDING AT 66650,
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_32
INDEX IN is_customer_32
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 66651 ENDING AT 68800,
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_33
INDEX IN is_customer_33
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 68801 ENDING AT 70950,
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_34
INDEX IN is_customer_34
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 70951 ENDING AT 73100,
  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_35
INDEX IN is_customer_35
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 73101 ENDING AT 75250,
  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_36
INDEX IN is_customer_36
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 75251 ENDING AT 77400,
  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_37
INDEX IN is_customer_37
ORGANIZE BY KEY SEQUENCE (
  C_ID STARTING FROM 1 ENDING AT 3000,
  C_W_ID STARTING FROM 77401 ENDING AT 79550,
  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_38
INDEX IN is_customer_38
ORGANIZE BY KEY SEQUENCE (
C_ID STARTING FROM 1 ENDING AT 3000,
C_W_ID STARTING FROM 79551 ENDING AT 81700,
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_01
INDEX IN ts_dist_01
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 1 ENDING AT 2150
)
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_02
INDEX IN ts_dist_02
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 2151 ENDING AT 4300
)
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_03
INDEX IN ts_dist_03
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 4301 ENDING AT 6450
)
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_04
INDEX IN ts_dist_04
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 6451 ENDING AT 8600
)
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_05
INDEX IN ts_dist_05
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 8601 ENDING AT 10750
)
)
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_06
INDEX IN ts_dist_06
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 10751 ENDING AT 12900
)
)
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_07
INDEX IN ts_dist_07
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 12901 ENDING AT 15050
)
)
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_08
INDEX IN ts_dist_08
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 15051 ENDING AT 17200
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT9;

CREATE TABLE DISTRICT9

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_09
INDEX IN ts_dist_09
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 17201 ENDING AT 19350
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT10;

CREATE TABLE DISTRICT10

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_10
INDEX IN ts_dist_10
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 19351 ENDING AT 21500
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT11;

CREATE TABLE DISTRICT11

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_11
INDEX IN ts_dist_11
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 21501 ENDING AT 23650
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT12;

CREATE TABLE DISTRICT12

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_12
INDEX IN ts_dist_12
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 23651 ENDING AT 25800
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT13;

CREATE TABLE DISTRICT13

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_13
INDEX IN ts_dist_13
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 25801 ENDING AT 27950
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT14;

CREATE TABLE DISTRICT14

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_14
INDEX IN ts_dist_14
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 27951 ENDING AT 30100
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT15;

CREATE TABLE DISTRICT15

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_15
INDEX IN ts_dist_15
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 30101 ENDING AT 32250
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT16;

CREATE TABLE DISTRICT16

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_16
INDEX IN ts_dist_16
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 32251 ENDING AT 34400
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT17;

CREATE TABLE DISTRICT17

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_17
INDEX IN ts_dist_17
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 34401 ENDING AT 36550
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT18;

CREATE TABLE DISTRICT18

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_18
INDEX IN ts_dist_18
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 36551 ENDING AT 38700
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT19;

CREATE TABLE DISTRICT19

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_19
INDEX IN ts_dist_19
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 38701 ENDING AT 40850
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT20;

CREATE TABLE DISTRICT20

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_20
INDEX IN ts_dist_20
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 40851 ENDING AT 43000
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT21;

CREATE TABLE DISTRICT21

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_21
INDEX IN ts_dist_21
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 43001 ENDING AT 45150
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT22;

CREATE TABLE DISTRICT22

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_22
INDEX IN ts_dist_22
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 45151 ENDING AT 47300
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT23;

CREATE TABLE DISTRICT23

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_23
INDEX IN ts_dist_23
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 47301 ENDING AT 49450
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT24;

CREATE TABLE DISTRICT24

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_24
INDEX IN ts_dist_24
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 49451 ENDING AT 51600
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT25;

CREATE TABLE DISTRICT25

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_25
INDEX IN ts_dist_25
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 51601 ENDING AT 53750
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT26;

CREATE TABLE DISTRICT26

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_26
INDEX IN ts_dist_26
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 53751 ENDING AT 55900
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT27;

CREATE TABLE DISTRICT27

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_27
INDEX IN ts_dist_27
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 55901 ENDING AT 58050
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT28;

CREATE TABLE DISTRICT28

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_28
INDEX IN ts_dist_28
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 58051 ENDING AT 60200
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT29;

CREATE TABLE DISTRICT29

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_29
INDEX IN ts_dist_29
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 60201 ENDING AT 62350
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT30;

CREATE TABLE DISTRICT30

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_30
INDEX IN ts_dist_30
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 62351 ENDING AT 64500
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT31;

CREATE TABLE DISTRICT31

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_31
INDEX IN ts_dist_31
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 64501 ENDING AT 66650
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT32;

CREATE TABLE DISTRICT32

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_32
INDEX IN ts_dist_32
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 66651 ENDING AT 68800
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT33;

CREATE TABLE DISTRICT33

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_33
INDEX IN ts_dist_33
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 68801 ENDING AT 70950
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT34;

CREATE TABLE DISTRICT34

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_34
INDEX IN ts_dist_34
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 70951 ENDING AT 73100
)
)
ALLOW OVERFLOW;

```

connect reset;

connect to TPCC in share mode;

DROP TABLE DISTRICT35;

CREATE TABLE DISTRICT35

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_35
INDEX IN ts_dist_35
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 73101 ENDING AT 75250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT36;
CREATE TABLE DISTRICT36

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_36
INDEX IN ts_dist_36
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 75251 ENDING AT 77400
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT37;
CREATE TABLE DISTRICT37

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_37
INDEX IN ts_dist_37
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 77401 ENDING AT 79550
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE DISTRICT38;
CREATE TABLE DISTRICT38

```

```

(
D_NEXT_O_ID INTEGER NOT NULL,
D_TAX 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_38
INDEX IN ts_dist_38
ORGANIZE BY KEY SEQUENCE (
D_ID STARTING FROM 1 ENDING AT 10,
D_W_ID STARTING FROM 79551 ENDING AT 81700
)
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_01
INDEX IN ts_history_01;

```

```
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_02
INDEX IN ts_history_02;

```

```
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_03
INDEX IN ts_history_03;

```

```
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_04
INDEX IN ts_history_04;
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_05
INDEX IN ts_history_05;
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_06
INDEX IN ts_history_06;
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_07
INDEX IN ts_history_07;
ALTER TABLE HISTORY7 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY8;
CREATE TABLE HISTORY8
(

```

```

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_08
INDEX IN ts_history_08;
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_09
INDEX IN ts_history_09;
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_10
INDEX IN ts_history_10;
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_11
INDEX IN ts_history_11;
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_12
INDEX IN ts_history_12;
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_13
INDEX IN ts_history_13;
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_14
INDEX IN ts_history_14;
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_15
INDEX IN ts_history_15;
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_16
INDEX IN ts_history_16;

```

```

ALTER TABLE HISTORY16 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY17;
CREATE TABLE HISTORY17
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE BIGINT NOT NULL,
H_AMOUNT INTEGER NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_17
INDEX IN ts_history_17;
ALTER TABLE HISTORY17 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY18;
CREATE TABLE HISTORY18
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE BIGINT NOT NULL,
H_AMOUNT INTEGER NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_18
INDEX IN ts_history_18;
ALTER TABLE HISTORY18 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY19;
CREATE TABLE HISTORY19
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE BIGINT NOT NULL,
H_AMOUNT INTEGER NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_19
INDEX IN ts_history_19;
ALTER TABLE HISTORY19 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY20;
CREATE TABLE HISTORY20
(
H_C_ID INTEGER NOT NULL,
H_C_D_ID SMALLINT NOT NULL,
H_C_W_ID INTEGER NOT NULL,
H_D_ID SMALLINT NOT NULL,
H_W_ID INTEGER NOT NULL,
H_DATE BIGINT NOT NULL,
H_AMOUNT INTEGER NOT NULL,
H_DATA CHAR(24) NOT NULL
)
IN ts_history_20
INDEX IN ts_history_20;
ALTER TABLE HISTORY20 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY21;

```

```

CREATE TABLE HISTORY21
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_21
INDEX IN ts_history_21;
ALTER TABLE HISTORY21 APPEND ON;
connect reset;

connect to TPCC in share mode;
DROP TABLE HISTORY22;
CREATE TABLE HISTORY22
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_22
INDEX IN ts_history_22;
ALTER TABLE HISTORY22 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY23;
CREATE TABLE HISTORY23
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_23
INDEX IN ts_history_23;
ALTER TABLE HISTORY23 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY24;
CREATE TABLE HISTORY24
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_24
INDEX IN ts_history_24;
ALTER TABLE HISTORY24 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY25;
CREATE TABLE HISTORY25
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_25
INDEX IN ts_history_25;
ALTER TABLE HISTORY25 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY26;
CREATE TABLE HISTORY26
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_26
INDEX IN ts_history_26;
ALTER TABLE HISTORY26 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY27;
CREATE TABLE HISTORY27
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_27
INDEX IN ts_history_27;
ALTER TABLE HISTORY27 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY28;
CREATE TABLE HISTORY28
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,
  H_W_ID      INTEGER NOT NULL,
  H_DATE      BIGINT NOT NULL,
  H_AMOUNT    INTEGER NOT NULL,
  H_DATA      CHAR(24) NOT NULL
)
IN ts_history_28
INDEX IN ts_history_28;
ALTER TABLE HISTORY28 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY29;
CREATE TABLE HISTORY29
(
  H_C_ID      INTEGER NOT NULL,
  H_C_D_ID    SMALLINT NOT NULL,
  H_C_W_ID    INTEGER NOT NULL,
  H_D_ID      SMALLINT NOT NULL,

```

```

        H_W_ID    INTEGER    NOT NULL,
        H_DATE    BIGINT    NOT NULL,
        H_AMOUNT  INTEGER    NOT NULL,
        H_DATA    CHAR(24)  NOT NULL
    )
    IN ts_history_29
    INDEX IN ts_history_29;
ALTER TABLE HISTORY29 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY30;
CREATE TABLE HISTORY30
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_30
INDEX IN ts_history_30;
ALTER TABLE HISTORY30 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY31;
CREATE TABLE HISTORY31
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_31
INDEX IN ts_history_31;
ALTER TABLE HISTORY31 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY32;
CREATE TABLE HISTORY32
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_32
INDEX IN ts_history_32;
ALTER TABLE HISTORY32 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY33;
CREATE TABLE HISTORY33
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL

```

```

    )
    IN ts_history_33
    INDEX IN ts_history_33;
ALTER TABLE HISTORY33 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY34;
CREATE TABLE HISTORY34
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_34
INDEX IN ts_history_34;
ALTER TABLE HISTORY34 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY35;
CREATE TABLE HISTORY35
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_35
INDEX IN ts_history_35;
ALTER TABLE HISTORY35 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY36;
CREATE TABLE HISTORY36
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_36
INDEX IN ts_history_36;
ALTER TABLE HISTORY36 APPEND ON;
connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY37;
CREATE TABLE HISTORY37
(
    H_C_ID    INTEGER    NOT NULL,
    H_C_D_ID  SMALLINT   NOT NULL,
    H_C_W_ID  INTEGER    NOT NULL,
    H_D_ID    SMALLINT   NOT NULL,
    H_W_ID    INTEGER    NOT NULL,
    H_DATE    BIGINT    NOT NULL,
    H_AMOUNT  INTEGER    NOT NULL,
    H_DATA    CHAR(24)  NOT NULL
)
IN ts_history_37
INDEX IN ts_history_37;
ALTER TABLE HISTORY37 APPEND ON;

```



```

connect reset;
connect to TPCC in share mode;
DROP TABLE HISTORY38;
CREATE TABLE HISTORY38
(
  H_C_ID    INTEGER NOT NULL,
  H_C_D_ID  SMALLINT NOT NULL,
  H_C_W_ID  INTEGER NOT NULL,
  H_D_ID    SMALLINT NOT NULL,
  H_W_ID    INTEGER NOT NULL,
  H_DATE    BIGINT  NOT NULL,
  H_AMOUNT  INTEGER NOT NULL,
  H_DATA    CHAR(24) NOT NULL
)
IN ts_history_38
INDEX IN ts_history_38;
ALTER TABLE HISTORY38 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_01
INDEX IN ts_item_01
ORGANIZE BY KEY SEQUENCE (
  I_ID STARTING FROM 1 ENDING AT 10000
)
ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;
connect reset;

```

CRTB_NEW_ORDER.ddl

```

connect to TPCC in share mode;
DROP TABLE NEW_ORDER1;
CREATE TABLE NEW_ORDER1
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_neword_01
INDEX IN ts_neword_01
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 1 ENDING AT 2150,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER2;
CREATE TABLE NEW_ORDER2
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_neword_02
INDEX IN ts_neword_02
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 2151 ENDING AT 4300,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER3;
CREATE TABLE NEW_ORDER3
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_neword_03
INDEX IN ts_neword_03
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 4301 ENDING AT 6450,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER4;
CREATE TABLE NEW_ORDER4
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_neword_04
INDEX IN ts_neword_04
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 6451 ENDING AT 8600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER5;
CREATE TABLE NEW_ORDER5
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_neword_05
INDEX IN ts_neword_05
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 8601 ENDING AT 10750,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER6;
CREATE TABLE NEW_ORDER6
(
  NO_O_ID   INTEGER NOT NULL,
  NO_D_ID   SMALLINT NOT NULL,
  NO_W_ID   INTEGER NOT NULL
)
IN ts_neword_06
INDEX IN ts_neword_06
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 10751 ENDING AT 12900,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER7;
CREATE TABLE NEW_ORDER7
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_07
INDEX IN ts_neword_07
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 12901 ENDING AT 15050,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER8;
CREATE TABLE NEW_ORDER8
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_08
INDEX IN ts_neword_08
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 15051 ENDING AT 17200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER9;
CREATE TABLE NEW_ORDER9
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_09
INDEX IN ts_neword_09
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 17201 ENDING AT 19350,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER10;
CREATE TABLE NEW_ORDER10
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_10
INDEX IN ts_neword_10
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 19351 ENDING AT 21500,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER11;

```

```

CREATE TABLE NEW_ORDER11
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_11
INDEX IN ts_neword_11
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 21501 ENDING AT 23650,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER12;
CREATE TABLE NEW_ORDER12
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_12
INDEX IN ts_neword_12
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 23651 ENDING AT 25800,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER13;
CREATE TABLE NEW_ORDER13
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_13
INDEX IN ts_neword_13
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 25801 ENDING AT 27950,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER14;
CREATE TABLE NEW_ORDER14
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_14
INDEX IN ts_neword_14
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 27951 ENDING AT 30100,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER15;
CREATE TABLE NEW_ORDER15
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,

```

```

NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_15
INDEX IN ts_neword_15
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 30101 ENDING AT 32250,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER16;
CREATE TABLE NEW_ORDER16
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_16
INDEX IN ts_neword_16

ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 32251 ENDING AT 34400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER17;
CREATE TABLE NEW_ORDER17
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_17
INDEX IN ts_neword_17
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 34401 ENDING AT 36550,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER18;
CREATE TABLE NEW_ORDER18
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_18
INDEX IN ts_neword_18
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 36551 ENDING AT 38700,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER19;
CREATE TABLE NEW_ORDER19
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL

```

```

)
)
IN ts_neword_19
INDEX IN ts_neword_19
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 38701 ENDING AT 40850,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER20;
CREATE TABLE NEW_ORDER20
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_20
INDEX IN ts_neword_20
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 40851 ENDING AT 43000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER21;
CREATE TABLE NEW_ORDER21
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_21
INDEX IN ts_neword_21
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 43001 ENDING AT 45150,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER22;
CREATE TABLE NEW_ORDER22
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_22
INDEX IN ts_neword_22
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 45151 ENDING AT 47300,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER23;
CREATE TABLE NEW_ORDER23
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
)
IN ts_neword_23
INDEX IN ts_neword_23
ORGANIZE BY KEY SEQUENCE (

```

```

NO_W_ID STARTING FROM 47301 ENDING AT 49450,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER24;
CREATE TABLE NEW_ORDER24
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_24
INDEX IN ts_neword_24
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 49451 ENDING AT 51600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER25;
CREATE TABLE NEW_ORDER25
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_25
INDEX IN ts_neword_25
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 51601 ENDING AT 53750,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER26;
CREATE TABLE NEW_ORDER26
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_26
INDEX IN ts_neword_26
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 53751 ENDING AT 55900,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER27;
CREATE TABLE NEW_ORDER27
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_27
INDEX IN ts_neword_27
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 55901 ENDING AT 58050,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER28;
CREATE TABLE NEW_ORDER28
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_28
INDEX IN ts_neword_28
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 58051 ENDING AT 60200,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER29;
CREATE TABLE NEW_ORDER29
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_29
INDEX IN ts_neword_29
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 60201 ENDING AT 62350,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER30;
CREATE TABLE NEW_ORDER30
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_30
INDEX IN ts_neword_30
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 62351 ENDING AT 64500,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER31;
CREATE TABLE NEW_ORDER31
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_31
INDEX IN ts_neword_31
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 64501 ENDING AT 66650,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER32;

```

```

CREATE TABLE NEW_ORDER32
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_32
INDEX IN ts_neword_32
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 66651 ENDING AT 68800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER33;
CREATE TABLE NEW_ORDER33
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_33
INDEX IN ts_neword_33
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 68801 ENDING AT 70950,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER34;
CREATE TABLE NEW_ORDER34
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_34
INDEX IN ts_neword_34
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 70951 ENDING AT 73100,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER35;
CREATE TABLE NEW_ORDER35
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_35
INDEX IN ts_neword_35
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 73101 ENDING AT 75250,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER36;
CREATE TABLE NEW_ORDER36
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,

```

```

  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_36
INDEX IN ts_neword_36
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 75251 ENDING AT 77400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER37;
CREATE TABLE NEW_ORDER37
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_37
INDEX IN ts_neword_37
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 77401 ENDING AT 79550,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER38;
CREATE TABLE NEW_ORDER38
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_38
INDEX IN ts_neword_38
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 79551 ENDING AT 81700,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 2101 ENDING AT 3675
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER46;
CREATE TABLE NEW_ORDER46
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_46
INDEX IN ts_neword_46
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 1 ENDING AT 2150,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER47;
CREATE TABLE NEW_ORDER47
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_47
INDEX IN ts_neword_47

```

```

ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 2151 ENDING AT 4300,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER48;
CREATE TABLE NEW_ORDER48
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_48
INDEX IN ts_neword_48
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 4301 ENDING AT 6450,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER49;
CREATE TABLE NEW_ORDER49
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_49
INDEX IN ts_neword_49
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 6451 ENDING AT 8600,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER50;
CREATE TABLE NEW_ORDER50
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_50
INDEX IN ts_neword_50
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 8601 ENDING AT 10750,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER51;
CREATE TABLE NEW_ORDER51
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_51
INDEX IN ts_neword_51
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 10751 ENDING AT 12900,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250

```

```

)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER52;
CREATE TABLE NEW_ORDER52
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_52
INDEX IN ts_neword_52
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 12901 ENDING AT 15050,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER53;
CREATE TABLE NEW_ORDER53
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_53
INDEX IN ts_neword_53
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 15051 ENDING AT 17200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER54;
CREATE TABLE NEW_ORDER54
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_54
INDEX IN ts_neword_54
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 17201 ENDING AT 19350,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER55;
CREATE TABLE NEW_ORDER55
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_55
INDEX IN ts_neword_55
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 19351 ENDING AT 21500,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;

```

```

DROP TABLE NEW_ORDER56;
CREATE TABLE NEW_ORDER56
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_56
INDEX IN ts_neword_56
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 21501 ENDING AT 23650,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER57;
CREATE TABLE NEW_ORDER57

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_57
INDEX IN ts_neword_57
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 23651 ENDING AT 25800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER58;
CREATE TABLE NEW_ORDER58

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_58
INDEX IN ts_neword_58
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 25801 ENDING AT 27950,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER59;
CREATE TABLE NEW_ORDER59

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_59
INDEX IN ts_neword_59
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 27951 ENDING AT 30100,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER60;
CREATE TABLE NEW_ORDER60

```

```

(
  NO_O_ID    INTEGER    NOT NULL,

```

```

  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_60
INDEX IN ts_neword_60
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 30101 ENDING AT 32250,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER61;
CREATE TABLE NEW_ORDER61

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_61
INDEX IN ts_neword_61
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 32251 ENDING AT 34400,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER62;
CREATE TABLE NEW_ORDER62

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_62
INDEX IN ts_neword_62
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 34401 ENDING AT 36550,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER63;
CREATE TABLE NEW_ORDER63

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_63
INDEX IN ts_neword_63
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 36551 ENDING AT 38700,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER64;
CREATE TABLE NEW_ORDER64

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_64

```

```

INDEX IN ts_neword_64
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 38701 ENDING AT 40850,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER65;
CREATE TABLE NEW_ORDER65
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_65
INDEX IN ts_neword_65
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 40851 ENDING AT 43000,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER66;
CREATE TABLE NEW_ORDER66
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_66
INDEX IN ts_neword_66
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 43001 ENDING AT 45150,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER67;
CREATE TABLE NEW_ORDER67
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_67
INDEX IN ts_neword_67
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 45151 ENDING AT 47300,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER68;
CREATE TABLE NEW_ORDER68
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_68
INDEX IN ts_neword_68
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 47301 ENDING AT 49450,
NO_D_ID STARTING FROM 1 ENDING AT 10,

```

```

NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER69;
CREATE TABLE NEW_ORDER69
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_69
INDEX IN ts_neword_69
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 49451 ENDING AT 51600,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER70;
CREATE TABLE NEW_ORDER70
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_70
INDEX IN ts_neword_70
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 51601 ENDING AT 53750,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER71;
CREATE TABLE NEW_ORDER71
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_71
INDEX IN ts_neword_71
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 53751 ENDING AT 55900,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER72;
CREATE TABLE NEW_ORDER72
(
NO_O_ID    INTEGER    NOT NULL,
NO_D_ID    SMALLINT   NOT NULL,
NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_72
INDEX IN ts_neword_72
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 55901 ENDING AT 58050,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```



```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER73;
CREATE TABLE NEW_ORDER73
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_73
INDEX IN ts_neword_73
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 58051 ENDING AT 60200,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER74;
CREATE TABLE NEW_ORDER74
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_74
INDEX IN ts_neword_74
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 60201 ENDING AT 62350,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER75;
CREATE TABLE NEW_ORDER75
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_75
INDEX IN ts_neword_75
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 62351 ENDING AT 64500,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER76;
CREATE TABLE NEW_ORDER76
(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_76
INDEX IN ts_neword_76
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 64501 ENDING AT 66650,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER77;
CREATE TABLE NEW_ORDER77

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_77
INDEX IN ts_neword_77
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 66651 ENDING AT 68800,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER78;
CREATE TABLE NEW_ORDER78

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_78
INDEX IN ts_neword_78
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 68801 ENDING AT 70950,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER79;
CREATE TABLE NEW_ORDER79

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_79
INDEX IN ts_neword_79
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 70951 ENDING AT 73100,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER80;
CREATE TABLE NEW_ORDER80

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL
)
IN ts_neword_80
INDEX IN ts_neword_80
ORGANIZE BY KEY SEQUENCE (
  NO_W_ID STARTING FROM 73101 ENDING AT 75250,
  NO_D_ID STARTING FROM 1 ENDING AT 10,
  NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER81;
CREATE TABLE NEW_ORDER81

```

```

(
  NO_O_ID    INTEGER    NOT NULL,
  NO_D_ID    SMALLINT   NOT NULL,
  NO_W_ID    INTEGER    NOT NULL

```

```

)
IN ts_neword_81
INDEX IN ts_neword_81
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 75251 ENDING AT 77400,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER82;
CREATE TABLE NEW_ORDER82
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_neword_82
INDEX IN ts_neword_82
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 77401 ENDING AT 79550,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDER83;
CREATE TABLE NEW_ORDER83
(
NO_O_ID INTEGER NOT NULL,
NO_D_ID SMALLINT NOT NULL,
NO_W_ID INTEGER NOT NULL
)
)
IN ts_neword_83
INDEX IN ts_neword_83
ORGANIZE BY KEY SEQUENCE (
NO_W_ID STARTING FROM 79551 ENDING AT 81700,
NO_D_ID STARTING FROM 1 ENDING AT 10,
NO_O_ID STARTING FROM 3676 ENDING AT 5250
)
)
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_01
INDEX IN is_order_01
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 1 ENDING AT 2150,
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_02
INDEX IN is_order_02
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 2151 ENDING AT 4300,
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_03
INDEX IN is_order_03
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 4301 ENDING AT 6450,
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_04
INDEX IN is_order_04
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 6451 ENDING AT 8600,
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_05
INDEX IN is_order_05
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 8601 ENDING AT 10750,
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_06
INDEX IN is_order_06
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 10751 ENDING AT 12900,
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_07
INDEX IN is_order_07
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 12901 ENDING AT 15050,
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_08

```

```

INDEX IN is_order_08
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 15051 ENDING AT 17200,
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_09
INDEX IN is_order_09
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 17201 ENDING AT 19350,
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_10
INDEX IN is_order_10
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 19351 ENDING AT 21500,
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_11
INDEX IN is_order_11
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 21501 ENDING AT 23650,
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_12
INDEX IN is_order_12
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 23651 ENDING AT 25800,
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_13
INDEX IN is_order_13
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 25801 ENDING AT 27950,
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_14
INDEX IN is_order_14
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 27951 ENDING AT 30100,
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_15
INDEX IN is_order_15
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 30101 ENDING AT 32250,
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_16
INDEX IN is_order_16
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 32251 ENDING AT 34400,
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_17
INDEX IN is_order_17
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 34401 ENDING AT 36550,
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_18
INDEX IN is_order_18
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 36551 ENDING AT 38700,
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_19
INDEX IN is_order_19
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 38701 ENDING AT 40850,
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_20
INDEX IN is_order_20
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 40851 ENDING AT 43000,
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_21

```

```

INDEX IN is_order_21
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 43001 ENDING AT 45150,
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_22
INDEX IN is_order_22
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 45151 ENDING AT 47300,
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_23
INDEX IN is_order_23
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 47301 ENDING AT 49450,
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_24
INDEX IN is_order_24
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 49451 ENDING AT 51600,
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_25
INDEX IN is_order_25
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 51601 ENDING AT 53750,
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_26
INDEX IN is_order_26
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 53751 ENDING AT 55900,
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_27
INDEX IN is_order_27
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 55901 ENDING AT 58050,
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_28
INDEX IN is_order_28
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 58051 ENDING AT 60200,
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_29
INDEX IN is_order_29
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 60201 ENDING AT 62350,
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_30
INDEX IN is_order_30
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 62351 ENDING AT 64500,
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_31
INDEX IN is_order_31
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 64501 ENDING AT 66650,
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_32
INDEX IN is_order_32
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 66651 ENDING AT 68800,
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_33
INDEX IN is_order_33
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 68801 ENDING AT 70950,
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_34
INDEX IN is_order_34
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 70951 ENDING AT 73100,
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_35
INDEX IN is_order_35
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 73101 ENDING AT 75250,
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_36
INDEX IN is_order_36
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 75251 ENDING AT 77400,
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_37
INDEX IN is_order_37
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,
O_W_ID STARTING FROM 77401 ENDING AT 79550,
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_38
INDEX IN is_order_38
ORGANIZE BY KEY SEQUENCE (
O_ID STARTING FROM 1 ENDING AT 3675,

O_W_ID STARTING FROM 79551 ENDING AT 81700,
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_01
INDEX IN ts_orderline_01
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 1 ENDING AT 2150,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE2;
CREATE TABLE ORDER_LINE2
(
OL_DELIVERY_D 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_02
INDEX IN ts_orderline_02

```

```

ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 2151 ENDING AT 4300,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE3;
CREATE TABLE ORDER_LINE3
(
OL_DELIVERY_D 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_03
INDEX IN ts_orderline_03
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 4301 ENDING AT 6450,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE4;
CREATE TABLE ORDER_LINE4
(
OL_DELIVERY_D 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_04
INDEX IN ts_orderline_04
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 6451 ENDING AT 8600,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE5;
CREATE TABLE ORDER_LINES
(
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_05
    INDEX IN ts_orderline_05
    ORGANIZE BY KEY SEQUENCE (
        OL_W_ID STARTING FROM 8601 ENDING AT 10750,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3675,
        OL_NUMBER STARTING FROM 1 ENDING AT 15
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE6;
CREATE TABLE ORDER_LINE6
(
    OL_DELIVERY_D 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_06
INDEX IN ts_orderline_06
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 10751 ENDING AT 12900,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE7;
CREATE TABLE ORDER_LINE7
(
    OL_DELIVERY_D 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_07
INDEX IN ts_orderline_07
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 12901 ENDING AT 15050,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE8;
CREATE TABLE ORDER_LINE8
(
    OL_DELIVERY_D 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_08
    INDEX IN ts_orderline_08
    ORGANIZE BY KEY SEQUENCE (
        OL_W_ID STARTING FROM 15051 ENDING AT 17200,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3675,
        OL_NUMBER STARTING FROM 1 ENDING AT 15
    )
    ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE9;
CREATE TABLE ORDER_LINE9
(
    OL_DELIVERY_D 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_09
INDEX IN ts_orderline_09
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 17201 ENDING AT 19350,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE10;
CREATE TABLE ORDER_LINE10
(
    OL_DELIVERY_D 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_10
INDEX IN ts_orderline_10
ORGANIZE BY KEY SEQUENCE (
    OL_W_ID STARTING FROM 19351 ENDING AT 21500,
    OL_D_ID STARTING FROM 1 ENDING AT 10,
    OL_O_ID STARTING FROM 1 ENDING AT 3675,
    OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE11;
CREATE TABLE ORDER_LINE11
(
    OL_DELIVERY_D 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_11
INDEX IN ts_orderline_11
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 21501 ENDING AT 23650,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE12;
CREATE TABLE ORDER_LINE12

```

```

(
OL_DELIVERY_D 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_12
INDEX IN ts_orderline_12
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 23651 ENDING AT 25800,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE13;
CREATE TABLE ORDER_LINE13

```

```

(
OL_DELIVERY_D 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_13
INDEX IN ts_orderline_13
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 25801 ENDING AT 27950,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE14;
CREATE TABLE ORDER_LINE14

```

```

(
OL_DELIVERY_D 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_14
INDEX IN ts_orderline_14
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 27951 ENDING AT 30100,
OL_D_ID STARTING FROM 1 ENDING AT 10,

```

```

OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE15;
CREATE TABLE ORDER_LINE15

```

```

(
OL_DELIVERY_D 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_15
INDEX IN ts_orderline_15
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 30101 ENDING AT 32250,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;

```

```

connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE16;
CREATE TABLE ORDER_LINE16

```

```

(
OL_DELIVERY_D 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_16
INDEX IN ts_orderline_16
ORGANIZE BY KEY SEQUENCE (
OL_W_ID STARTING FROM 32251 ENDING AT 34400,
OL_D_ID STARTING FROM 1 ENDING AT 10,
OL_O_ID STARTING FROM 1 ENDING AT 3675,
OL_NUMBER STARTING FROM 1 ENDING AT 15
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE17;
CREATE TABLE ORDER_LINE17
(
  OL_DELIVERY_D 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_17
INDEX IN ts_orderline_17
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 34401 ENDING AT 36550,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE18;
CREATE TABLE ORDER_LINE18
(
  OL_DELIVERY_D 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_18
INDEX IN ts_orderline_18
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 36551 ENDING AT 38700,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE19;
CREATE TABLE ORDER_LINE19
(
  OL_DELIVERY_D 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_19
INDEX IN ts_orderline_19
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 38701 ENDING AT 40850,
  OL_D_ID STARTING FROM 1 ENDING AT 10,

```

```

  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE20;
CREATE TABLE ORDER_LINE20
(
  OL_DELIVERY_D 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_20
INDEX IN ts_orderline_20
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 40851 ENDING AT 43000,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE21;
CREATE TABLE ORDER_LINE21
(
  OL_DELIVERY_D 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_21
INDEX IN ts_orderline_21
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 43001 ENDING AT 45150,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE22;
CREATE TABLE ORDER_LINE22
(
  OL_DELIVERY_D 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_22
INDEX IN ts_orderline_22

```

```

ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 45151 ENDING AT 47300,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE23;
CREATE TABLE ORDER_LINE23
(
  OL_DELIVERY_D 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_23
INDEX IN ts_orderline_23
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 47301 ENDING AT 49450,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE24;
CREATE TABLE ORDER_LINE24
(
  OL_DELIVERY_D 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_24
INDEX IN ts_orderline_24
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 49451 ENDING AT 51600,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
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_25
INDEX IN ts_orderline_25
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 51601 ENDING AT 53750,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE26;
CREATE TABLE ORDER_LINE26
(
  OL_DELIVERY_D 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_26
INDEX IN ts_orderline_26
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 53751 ENDING AT 55900,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE27;
CREATE TABLE ORDER_LINE27
(
  OL_DELIVERY_D 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_27
INDEX IN ts_orderline_27
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 55901 ENDING AT 58050,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE28;
CREATE TABLE ORDER_LINE28
(
  OL_DELIVERY_D 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_28
INDEX IN ts_orderline_28
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 58051 ENDING AT 60200,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE29;
CREATE TABLE ORDER_LINE29
(
  OL_DELIVERY_D 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_29
INDEX IN ts_orderline_29
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 60201 ENDING AT 62350,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE30;
CREATE TABLE ORDER_LINE30
(
  OL_DELIVERY_D 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_30
INDEX IN ts_orderline_30
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 62351 ENDING AT 64500,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE31;
CREATE TABLE ORDER_LINE31
(
  OL_DELIVERY_D 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_31
INDEX IN ts_orderline_31
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 64501 ENDING AT 66650,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE32;
CREATE TABLE ORDER_LINE32
(
  OL_DELIVERY_D 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_32
INDEX IN ts_orderline_32
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 66651 ENDING AT 68800,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE33;
CREATE TABLE ORDER_LINE33
(
  OL_DELIVERY_D 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_33
INDEX IN ts_orderline_33
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 68801 ENDING AT 70950,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE34;
CREATE TABLE ORDER_LINE34

```

```
(
  OL_DELIVERY_D 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_34
INDEX IN ts_orderline_34
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 70951 ENDING AT 73100,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
```

connect reset;

connect to TPCC in share mode;

DROP TABLE ORDER_LINE35;

CREATE TABLE ORDER_LINE35

```
(
  OL_DELIVERY_D 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_35
INDEX IN ts_orderline_35
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 73101 ENDING AT 75250,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
```

connect reset;

connect to TPCC in share mode;

DROP TABLE ORDER_LINE36;

CREATE TABLE ORDER_LINE36

```
(
  OL_DELIVERY_D 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_36
INDEX IN ts_orderline_36
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 75251 ENDING AT 77400,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
```

connect reset;

connect to TPCC in share mode;

DROP TABLE ORDER_LINE37;

CREATE TABLE ORDER_LINE37

```
(
  OL_DELIVERY_D 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_37
INDEX IN ts_orderline_37
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 77401 ENDING AT 79550,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
```

connect reset;

connect to TPCC in share mode;

DROP TABLE ORDER_LINE38;

CREATE TABLE ORDER_LINE38

```
(
  OL_DELIVERY_D 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_38
INDEX IN ts_orderline_38
ORGANIZE BY KEY SEQUENCE (
  OL_W_ID STARTING FROM 79551 ENDING AT 81700,
  OL_D_ID STARTING FROM 1 ENDING AT 10,
  OL_O_ID STARTING FROM 1 ENDING AT 3675,
  OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
```

connect reset;

CRTB_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_01
INDEX IN ts_stock_01
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 1 ENDING AT 1075
)
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_02
INDEX IN ts_stock_02
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 1076 ENDING AT 2150
)
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_03
INDEX IN ts_stock_03
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 2151 ENDING AT 3225
)
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_04
INDEX IN ts_stock_04
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 3226 ENDING AT 4300
)
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_05
INDEX IN ts_stock_05
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 4301 ENDING AT 5375
)
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_06
INDEX IN ts_stock_06
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 5376 ENDING AT 6450
)
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_07
INDEX IN ts_stock_07
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 6451 ENDING AT 7525
)
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_08

```

```

INDEX IN ts_stock_08
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 7526 ENDING AT 8600
)
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_09
INDEX IN ts_stock_09
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 8601 ENDING AT 9675
)
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_10
INDEX IN ts_stock_10
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 9676 ENDING AT 10750
)
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_11
INDEX IN ts_stock_11
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 10751 ENDING AT 11825
)
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_12
INDEX IN ts_stock_12
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 11826 ENDING AT 12900
)
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_13
INDEX IN ts_stock_13
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 12901 ENDING AT 13975
)
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_14
INDEX IN ts_stock_14
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 13976 ENDING AT 15050
)
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_15
INDEX IN ts_stock_15
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 15051 ENDING AT 16125
)
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_16
INDEX IN ts_stock_16
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 16126 ENDING AT 17200
)
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_17
INDEX IN ts_stock_17
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 17201 ENDING AT 18275
)
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_18
INDEX IN ts_stock_18
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 18276 ENDING AT 19350
)
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_19
INDEX IN ts_stock_19
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 19351 ENDING AT 20425
)
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_20

```

```

INDEX IN ts_stock_20
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 20426 ENDING AT 21500
)
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_21
INDEX IN ts_stock_21
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 21501 ENDING AT 22575
)
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_22
INDEX IN ts_stock_22
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 22576 ENDING AT 23650
)
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_23
INDEX IN ts_stock_23
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 23651 ENDING AT 24725
)
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_24
INDEX IN ts_stock_24
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 24726 ENDING AT 25800
)
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_25
INDEX IN ts_stock_25
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 25801 ENDING AT 26875
)
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_26
INDEX IN ts_stock_26
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 26876 ENDING AT 27950
)
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_27
INDEX IN ts_stock_27
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 27951 ENDING AT 29025
)
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_28
INDEX IN ts_stock_28
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 29026 ENDING AT 30100
)
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_29
INDEX IN ts_stock_29
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 30101 ENDING AT 31175
)
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_30
INDEX IN ts_stock_30
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 31176 ENDING AT 32250
)
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_31
INDEX IN ts_stock_31
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 32251 ENDING AT 33325
)
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_32

```

```

INDEX IN ts_stock_32
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 33326 ENDING AT 34400
)
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_33
INDEX IN ts_stock_33
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 34401 ENDING AT 35475
)
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_34
INDEX IN ts_stock_34
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 35476 ENDING AT 36550
)
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_35
INDEX IN ts_stock_35
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 36551 ENDING AT 37625
)
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_36
INDEX IN ts_stock_36
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 37626 ENDING AT 38700
)
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_37
INDEX IN ts_stock_37
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 38701 ENDING AT 39775
)
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_38
INDEX IN ts_stock_38
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 39776 ENDING AT 40850
)
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_39
INDEX IN ts_stock_39
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 40851 ENDING AT 41925

```

```

)
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_40
INDEX IN ts_stock_40
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 41926 ENDING AT 43000
)
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_41
INDEX IN ts_stock_41
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 43001 ENDING AT 44075
)
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_42
INDEX IN ts_stock_42
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 44076 ENDING AT 45150
)
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_43
INDEX IN ts_stock_43
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 45151 ENDING AT 46225
)
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_44
INDEX IN ts_stock_44
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 46226 ENDING AT 47300
)
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_45

INDEX IN ts_stock_45

```

ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 47301 ENDING AT 48375
)

```

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_46

INDEX IN ts_stock_46

```

ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 48376 ENDING AT 49450
)

```

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_47

INDEX IN ts_stock_47

```

ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 49451 ENDING AT 50525
)

```

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_48

INDEX IN ts_stock_48

```

ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 50526 ENDING AT 51600
)

```

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_49
INDEX IN ts_stock_49
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 51601 ENDING AT 52675
)
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_50
INDEX IN ts_stock_50
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 52676 ENDING AT 53750
)
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_51
INDEX IN ts_stock_51
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,

```

```

S_W_ID STARTING FROM 53751 ENDING AT 54825
)
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_52
INDEX IN ts_stock_52
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 54826 ENDING AT 55900
)
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_53
INDEX IN ts_stock_53
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 55901 ENDING AT 56975
)
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_54
INDEX IN ts_stock_54
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 56976 ENDING AT 58050
)
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_55
INDEX IN ts_stock_55
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 58051 ENDING AT 59125
)
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_56
INDEX IN ts_stock_56
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 59126 ENDING AT 60200
)
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_57
INDEX IN ts_stock_57
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 60201 ENDING AT 61275
)
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_58
INDEX IN ts_stock_58
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 61276 ENDING AT 62350
)
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_59
INDEX IN ts_stock_59
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 62351 ENDING AT 63425
)
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_60
INDEX IN ts_stock_60
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 63426 ENDING AT 64500
)
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_61
INDEX IN ts_stock_61
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 64501 ENDING AT 65575
)
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_62
INDEX IN ts_stock_62
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 100000,
  S_W_ID STARTING FROM 65576 ENDING AT 66650
)
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_63

```

```

INDEX IN ts_stock_63
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 66651 ENDING AT 67725
)
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_64
INDEX IN ts_stock_64
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 67726 ENDING AT 68800
)
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_65
INDEX IN ts_stock_65
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 68801 ENDING AT 69875
)
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_66
INDEX IN ts_stock_66
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 69876 ENDING AT 70950
)
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_67
INDEX IN ts_stock_67
ORGANIZE BY KEY SEQUENCE (
  S_I_ID STARTING FROM 1 ENDING AT 10000,
  S_W_ID STARTING FROM 70951 ENDING AT 72025
)
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_68
INDEX IN ts_stock_68
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 72026 ENDING AT 73100
)
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_69
INDEX IN ts_stock_69
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 73101 ENDING AT 74175
)
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_70
INDEX IN ts_stock_70
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 74176 ENDING AT 75250
)
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_71
INDEX IN ts_stock_71
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 75251 ENDING AT 76325
)
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_72
INDEX IN ts_stock_72
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 100000,
S_W_ID STARTING FROM 76326 ENDING AT 77400
)
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_73
INDEX IN ts_stock_73
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 77401 ENDING AT 78475
)
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_74
INDEX IN ts_stock_74
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 78476 ENDING AT 79550
)
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_75

```

```

INDEX IN ts_stock_75
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 79551 ENDING AT 80625
)
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_76
INDEX IN ts_stock_76
ORGANIZE BY KEY SEQUENCE (
S_I_ID STARTING FROM 1 ENDING AT 10000,
S_W_ID STARTING FROM 80626 ENDING AT 81700
)
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_01
INDEX IN ts_ware_01
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 1 ENDING AT 2150
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE2;
CREATE TABLE WAREHOUSE2
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,

```

```

W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_02
INDEX IN ts_ware_02
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 2151 ENDING AT 4300
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE3;
CREATE TABLE WAREHOUSE3
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_03
INDEX IN ts_ware_03
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 4301 ENDING AT 6450
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE4;
CREATE TABLE WAREHOUSE4
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_04
INDEX IN ts_ware_04
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 6451 ENDING AT 8600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE5;
CREATE TABLE WAREHOUSE5
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_05
INDEX IN ts_ware_05
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 8601 ENDING AT 10750
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE6;
CREATE TABLE WAREHOUSE6
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_06
INDEX IN ts_ware_06
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 10751 ENDING AT 12900
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE7;
CREATE TABLE WAREHOUSE7
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_07
INDEX IN ts_ware_07
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 12901 ENDING AT 15050
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE8;
CREATE TABLE WAREHOUSE8
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_08
INDEX IN ts_ware_08
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 15051 ENDING AT 17200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE9;
CREATE TABLE WAREHOUSE9
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,

```

```

W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_09
INDEX IN ts_ware_09
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 17201 ENDING AT 19350
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE10;
CREATE TABLE WAREHOUSE10
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_10
INDEX IN ts_ware_10
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 19351 ENDING AT 21500
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE11;
CREATE TABLE WAREHOUSE11
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_11
INDEX IN ts_ware_11
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 21501 ENDING AT 23650
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE12;
CREATE TABLE WAREHOUSE12
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_12

```

```

INDEX IN ts_ware_12
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 23651 ENDING AT 25800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE13;
CREATE TABLE WAREHOUSE13
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_13
INDEX IN ts_ware_13
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 25801 ENDING AT 27950
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE14;
CREATE TABLE WAREHOUSE14
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_14
INDEX IN ts_ware_14
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 27951 ENDING AT 30100
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE15;
CREATE TABLE WAREHOUSE15
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_15
INDEX IN ts_ware_15
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 30101 ENDING AT 32250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE16;
CREATE TABLE WAREHOUSE16

```



```

(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,

W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_16
INDEX IN ts_ware_16
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 32251 ENDING AT 34400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE17;
CREATE TABLE WAREHOUSE17
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_17
INDEX IN ts_ware_17
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 34401 ENDING AT 36550
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE18;
CREATE TABLE WAREHOUSE18
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_18
INDEX IN ts_ware_18
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 36551 ENDING AT 38700
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE19;
CREATE TABLE WAREHOUSE19
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,

```

```

W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_19
INDEX IN ts_ware_19
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 38701 ENDING AT 40850
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE20;
CREATE TABLE WAREHOUSE20
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_20
INDEX IN ts_ware_20
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 40851 ENDING AT 43000
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE21;
CREATE TABLE WAREHOUSE21
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_21
INDEX IN ts_ware_21
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 43001 ENDING AT 45150
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE22;
CREATE TABLE WAREHOUSE22
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_22
INDEX IN ts_ware_22
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 45151 ENDING AT 47300
)

```

```

ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE23;
CREATE TABLE WAREHOUSE23
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_23
INDEX IN ts_ware_23
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 47301 ENDING AT 49450
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE24;
CREATE TABLE WAREHOUSE24
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_24
INDEX IN ts_ware_24
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 49451 ENDING AT 51600
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE25;
CREATE TABLE WAREHOUSE25
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_25
INDEX IN ts_ware_25
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 51601 ENDING AT 53750
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE26;
CREATE TABLE WAREHOUSE26
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,

```

```

  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_26
INDEX IN ts_ware_26
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 53751 ENDING AT 55900
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE27;
CREATE TABLE WAREHOUSE27
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_27
INDEX IN ts_ware_27
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 55901 ENDING AT 58050
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE28;
CREATE TABLE WAREHOUSE28
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_28
INDEX IN ts_ware_28
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 58051 ENDING AT 60200
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE29;
CREATE TABLE WAREHOUSE29
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_29
INDEX IN ts_ware_29

```

```

ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 60201 ENDING AT 62350
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE30;
CREATE TABLE WAREHOUSE30
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_30
INDEX IN ts_ware_30
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 62351 ENDING AT 64500
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE31;
CREATE TABLE WAREHOUSE31
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_31
INDEX IN ts_ware_31
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 64501 ENDING AT 66650
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE32;
CREATE TABLE WAREHOUSE32
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_32
INDEX IN ts_ware_32
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 66651 ENDING AT 68800
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE33;
CREATE TABLE WAREHOUSE33
(

```

```

  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_33
INDEX IN ts_ware_33
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 68801 ENDING AT 70950
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE34;
CREATE TABLE WAREHOUSE34
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_34
INDEX IN ts_ware_34
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 70951 ENDING AT 73100
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE35;
CREATE TABLE WAREHOUSE35
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)
IN ts_ware_35
INDEX IN ts_ware_35
ORGANIZE BY KEY SEQUENCE (
  W_ID STARTING FROM 73101 ENDING AT 75250
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE36;
CREATE TABLE WAREHOUSE36
(
  W_NAME CHAR(10) NOT NULL,
  W_STREET_1 CHAR(20) NOT NULL,
  W_STREET_2 CHAR(20) NOT NULL,
  W_CITY CHAR(20) NOT NULL,
  W_STATE CHAR(2) NOT NULL,
  W_ZIP CHAR(9) NOT NULL,
  W_TAX INTEGER NOT NULL,
  W_YTD BIGINT NOT NULL,
  W_ID INTEGER NOT NULL
)

```

```

)
IN ts_ware_36
INDEX IN ts_ware_36
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 75251 ENDING AT 77400
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE37;
CREATE TABLE WAREHOUSE37
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_37
INDEX IN ts_ware_37
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 77401 ENDING AT 79550
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE WAREHOUSE38;
CREATE TABLE WAREHOUSE38
(
W_NAME CHAR(10) NOT NULL,
W_STREET_1 CHAR(20) NOT NULL,
W_STREET_2 CHAR(20) NOT NULL,
W_CITY CHAR(20) NOT NULL,
W_STATE CHAR(2) NOT NULL,
W_ZIP CHAR(9) NOT NULL,
W_TAX INTEGER NOT NULL,
W_YTD BIGINT NOT NULL,
W_ID INTEGER NOT NULL
)
IN ts_ware_38
INDEX IN ts_ware_38
ORGANIZE BY KEY SEQUENCE (
W_ID STARTING FROM 79551 ENDING AT 81700
)
ALLOW OVERFLOW;
connect reset;

```

CRVW CUSTOMER.ddl

```

connect to TPCC in share mode;
DROP VIEW CUSTOMER;
CREATE VIEW CUSTOMER
(C_ID,
C_STATE,
C_ZIP,
C_PHONE,
C_SINCE,
C_CREDIT_LIM,
C_MIDDLE,
C_CREDIT,
C_DISCOUNT,
C_DATA,
C_LAST,
C_FIRST,
C_STREET_1,
C_STREET_2,
C_CITY,

```

```

C_D_ID,
C_W_ID,
C_DELIVERY_CNT,
C_BALANCE,
C_YTD_PAYMENT,
C_PAYMENT_CNT
) AS SELECT * FROM CUSTOMER1 UNION ALL
SELECT * FROM CUSTOMER2 UNION ALL
SELECT * FROM CUSTOMER3 UNION ALL
SELECT * FROM CUSTOMER4 UNION ALL
SELECT * FROM CUSTOMER5 UNION ALL
SELECT * FROM CUSTOMER6 UNION ALL
SELECT * FROM CUSTOMER7 UNION ALL
SELECT * FROM CUSTOMER8 UNION ALL
SELECT * FROM CUSTOMER9 UNION ALL
SELECT * FROM CUSTOMER10 UNION ALL
SELECT * FROM CUSTOMER11 UNION ALL
SELECT * FROM CUSTOMER12 UNION ALL
SELECT * FROM CUSTOMER13 UNION ALL
SELECT * FROM CUSTOMER14 UNION ALL
SELECT * FROM CUSTOMER15 UNION ALL
SELECT * FROM CUSTOMER16 UNION ALL
SELECT * FROM CUSTOMER17 UNION ALL
SELECT * FROM CUSTOMER18 UNION ALL
SELECT * FROM CUSTOMER19 UNION ALL
SELECT * FROM CUSTOMER20 UNION ALL
SELECT * FROM CUSTOMER21 UNION ALL
SELECT * FROM CUSTOMER22 UNION ALL
SELECT * FROM CUSTOMER23 UNION ALL
SELECT * FROM CUSTOMER24 UNION ALL
SELECT * FROM CUSTOMER25 UNION ALL
SELECT * FROM CUSTOMER26 UNION ALL
SELECT * FROM CUSTOMER27 UNION ALL
SELECT * FROM CUSTOMER28 UNION ALL
SELECT * FROM CUSTOMER29 UNION ALL
SELECT * FROM CUSTOMER30 UNION ALL
SELECT * FROM CUSTOMER31 UNION ALL
SELECT * FROM CUSTOMER32 UNION ALL
SELECT * FROM CUSTOMER33 UNION ALL
SELECT * FROM CUSTOMER34 UNION ALL
SELECT * FROM CUSTOMER35 UNION ALL
SELECT * FROM CUSTOMER36 UNION ALL
SELECT * FROM CUSTOMER37 UNION ALL
SELECT * FROM CUSTOMER38
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW DISTRICT.ddl

```

connect to TPCC in share mode;
DROP VIEW DISTRICT;
CREATE VIEW DISTRICT
(D_NEXT_O_ID,
D_TAX,
D_YTD,
D_NAME,
D_STREET_1,
D_STREET_2,
D_CITY,
D_STATE,
D_ZIP,
D_ID,
D_W_ID
) AS SELECT * FROM DISTRICT1 UNION ALL
SELECT * FROM DISTRICT2 UNION ALL
SELECT * FROM DISTRICT3 UNION ALL
SELECT * FROM DISTRICT4 UNION ALL
SELECT * FROM DISTRICT5 UNION ALL
SELECT * FROM DISTRICT6 UNION ALL
SELECT * FROM DISTRICT7 UNION ALL

```

```

SELECT * FROM DISTRICT8 UNION ALL
SELECT * FROM DISTRICT9 UNION ALL
SELECT * FROM DISTRICT10 UNION ALL
SELECT * FROM DISTRICT11 UNION ALL
SELECT * FROM DISTRICT12 UNION ALL
SELECT * FROM DISTRICT13 UNION ALL
SELECT * FROM DISTRICT14 UNION ALL
SELECT * FROM DISTRICT15 UNION ALL
SELECT * FROM DISTRICT16 UNION ALL
SELECT * FROM DISTRICT17 UNION ALL
SELECT * FROM DISTRICT18 UNION ALL
SELECT * FROM DISTRICT19 UNION ALL
SELECT * FROM DISTRICT20 UNION ALL

```

```

SELECT * FROM DISTRICT21 UNION ALL
SELECT * FROM DISTRICT22 UNION ALL
SELECT * FROM DISTRICT23 UNION ALL
SELECT * FROM DISTRICT24 UNION ALL
SELECT * FROM DISTRICT25 UNION ALL
SELECT * FROM DISTRICT26 UNION ALL
SELECT * FROM DISTRICT27 UNION ALL
SELECT * FROM DISTRICT28 UNION ALL
SELECT * FROM DISTRICT29 UNION ALL
SELECT * FROM DISTRICT30 UNION ALL
SELECT * FROM DISTRICT31 UNION ALL
SELECT * FROM DISTRICT32 UNION ALL
SELECT * FROM DISTRICT33 UNION ALL
SELECT * FROM DISTRICT34 UNION ALL
SELECT * FROM DISTRICT35 UNION ALL
SELECT * FROM DISTRICT36 UNION ALL
SELECT * FROM DISTRICT37 UNION ALL
SELECT * FROM DISTRICT38
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW_HISTORY.ddl

```

connect to TPCC in share mode;
DROP VIEW HISTORY;
CREATE VIEW HISTORY
(H_C_ID,
H_C_D_ID,
H_C_W_ID,
H_D_ID,
H_W_ID,
H_DATE,
H_AMOUNT,
H_DATA
) AS SELECT * FROM HISTORY1 UNION ALL
SELECT * FROM HISTORY2 UNION ALL
SELECT * FROM HISTORY3 UNION ALL
SELECT * FROM HISTORY4 UNION ALL
SELECT * FROM HISTORY5 UNION ALL
SELECT * FROM HISTORY6 UNION ALL
SELECT * FROM HISTORY7 UNION ALL
SELECT * FROM HISTORY8 UNION ALL
SELECT * FROM HISTORY9 UNION ALL
SELECT * FROM HISTORY10 UNION ALL
SELECT * FROM HISTORY11 UNION ALL
SELECT * FROM HISTORY12 UNION ALL
SELECT * FROM HISTORY13 UNION ALL
SELECT * FROM HISTORY14 UNION ALL
SELECT * FROM HISTORY15 UNION ALL
SELECT * FROM HISTORY16 UNION ALL
SELECT * FROM HISTORY17 UNION ALL
SELECT * FROM HISTORY18 UNION ALL
SELECT * FROM HISTORY19 UNION ALL
SELECT * FROM HISTORY20 UNION ALL
SELECT * FROM HISTORY21 UNION ALL
SELECT * FROM HISTORY22 UNION ALL

```

```

SELECT * FROM HISTORY23 UNION ALL
SELECT * FROM HISTORY24 UNION ALL
SELECT * FROM HISTORY25 UNION ALL
SELECT * FROM HISTORY26 UNION ALL
SELECT * FROM HISTORY27 UNION ALL
SELECT * FROM HISTORY28 UNION ALL
SELECT * FROM HISTORY29 UNION ALL
SELECT * FROM HISTORY30 UNION ALL
SELECT * FROM HISTORY31 UNION ALL
SELECT * FROM HISTORY32 UNION ALL
SELECT * FROM HISTORY33 UNION ALL
SELECT * FROM HISTORY34 UNION ALL
SELECT * FROM HISTORY35 UNION ALL
SELECT * FROM HISTORY36 UNION ALL
SELECT * FROM HISTORY37 UNION ALL
SELECT * FROM HISTORY38
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_ORDER1 UNION ALL
SELECT * FROM NEW_ORDER2 UNION ALL
SELECT * FROM NEW_ORDER3 UNION ALL
SELECT * FROM NEW_ORDER4 UNION ALL
SELECT * FROM NEW_ORDER5 UNION ALL
SELECT * FROM NEW_ORDER6 UNION ALL
SELECT * FROM NEW_ORDER7 UNION ALL
SELECT * FROM NEW_ORDER8 UNION ALL
SELECT * FROM NEW_ORDER9 UNION ALL
SELECT * FROM NEW_ORDER10 UNION ALL
SELECT * FROM NEW_ORDER11 UNION ALL
SELECT * FROM NEW_ORDER12 UNION ALL
SELECT * FROM NEW_ORDER13 UNION ALL
SELECT * FROM NEW_ORDER14 UNION ALL
SELECT * FROM NEW_ORDER15 UNION ALL
SELECT * FROM NEW_ORDER16 UNION ALL
SELECT * FROM NEW_ORDER17 UNION ALL
SELECT * FROM NEW_ORDER18 UNION ALL
SELECT * FROM NEW_ORDER19 UNION ALL
SELECT * FROM NEW_ORDER20 UNION ALL
SELECT * FROM NEW_ORDER21 UNION ALL
SELECT * FROM NEW_ORDER22 UNION ALL
SELECT * FROM NEW_ORDER23 UNION ALL
SELECT * FROM NEW_ORDER24 UNION ALL
SELECT * FROM NEW_ORDER25 UNION ALL
SELECT * FROM NEW_ORDER26 UNION ALL
SELECT * FROM NEW_ORDER27 UNION ALL
SELECT * FROM NEW_ORDER28 UNION ALL
SELECT * FROM NEW_ORDER29 UNION ALL
SELECT * FROM NEW_ORDER30 UNION ALL
SELECT * FROM NEW_ORDER31 UNION ALL
SELECT * FROM NEW_ORDER32 UNION ALL
SELECT * FROM NEW_ORDER33 UNION ALL
SELECT * FROM NEW_ORDER34 UNION ALL
SELECT * FROM NEW_ORDER35 UNION ALL
SELECT * FROM NEW_ORDER36 UNION ALL
SELECT * FROM NEW_ORDER37 UNION ALL
SELECT * FROM NEW_ORDER38 UNION ALL
SELECT * FROM NEW_ORDER46 UNION ALL
SELECT * FROM NEW_ORDER47 UNION ALL
SELECT * FROM NEW_ORDER48 UNION ALL
SELECT * FROM NEW_ORDER49 UNION ALL
SELECT * FROM NEW_ORDER50 UNION ALL

```

```

SELECT * FROM NEW_ORDER51 UNION ALL
SELECT * FROM NEW_ORDER52 UNION ALL
SELECT * FROM NEW_ORDER53 UNION ALL
SELECT * FROM NEW_ORDER54 UNION ALL
SELECT * FROM NEW_ORDER55 UNION ALL
SELECT * FROM NEW_ORDER56 UNION ALL
SELECT * FROM NEW_ORDER57 UNION ALL
SELECT * FROM NEW_ORDER58 UNION ALL
SELECT * FROM NEW_ORDER59 UNION ALL
SELECT * FROM NEW_ORDER60 UNION ALL
SELECT * FROM NEW_ORDER61 UNION ALL
SELECT * FROM NEW_ORDER62 UNION ALL
SELECT * FROM NEW_ORDER63 UNION ALL
SELECT * FROM NEW_ORDER64 UNION ALL
SELECT * FROM NEW_ORDER65 UNION ALL
SELECT * FROM NEW_ORDER66 UNION ALL
SELECT * FROM NEW_ORDER67 UNION ALL
SELECT * FROM NEW_ORDER68 UNION ALL
SELECT * FROM NEW_ORDER69 UNION ALL
SELECT * FROM NEW_ORDER70 UNION ALL
SELECT * FROM NEW_ORDER71 UNION ALL
SELECT * FROM NEW_ORDER72 UNION ALL
SELECT * FROM NEW_ORDER73 UNION ALL
SELECT * FROM NEW_ORDER74 UNION ALL
SELECT * FROM NEW_ORDER75 UNION ALL
SELECT * FROM NEW_ORDER76 UNION ALL
SELECT * FROM NEW_ORDER77 UNION ALL
SELECT * FROM NEW_ORDER78 UNION ALL
SELECT * FROM NEW_ORDER79 UNION ALL
SELECT * FROM NEW_ORDER80 UNION ALL
SELECT * FROM NEW_ORDER81 UNION ALL
SELECT * FROM NEW_ORDER82 UNION ALL
SELECT * FROM NEW_ORDER83
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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW ORDER_LINE.ddl

```

connect to TPCC in share mode;
DROP VIEW ORDER_LINE;
CREATE VIEW ORDER_LINE
(OL_DELIVERY_D,
 OL_AMOUNT,
 OL_I_ID,
 OL_SUPPLY_W_ID,
 OL_QUANTITY,
 OL_DIST_INFO,
 OL_O_ID,
 OL_D_ID,
 OL_W_ID,
 OL_NUMBER
) AS SELECT * FROM ORDER_LINE1 UNION ALL
SELECT * FROM ORDER_LINE2 UNION ALL
SELECT * FROM ORDER_LINE3 UNION ALL
SELECT * FROM ORDER_LINE4 UNION ALL
SELECT * FROM ORDER_LINE5 UNION ALL
SELECT * FROM ORDER_LINE6 UNION ALL
SELECT * FROM ORDER_LINE7 UNION ALL
SELECT * FROM ORDER_LINE8 UNION ALL
SELECT * FROM ORDER_LINE9 UNION ALL
SELECT * FROM ORDER_LINE10 UNION ALL
SELECT * FROM ORDER_LINE11 UNION ALL
SELECT * FROM ORDER_LINE12 UNION ALL
SELECT * FROM ORDER_LINE13 UNION ALL
SELECT * FROM ORDER_LINE14 UNION ALL
SELECT * FROM ORDER_LINE15 UNION ALL
SELECT * FROM ORDER_LINE16 UNION ALL
SELECT * FROM ORDER_LINE17 UNION ALL
SELECT * FROM ORDER_LINE18 UNION ALL
SELECT * FROM ORDER_LINE19 UNION ALL
SELECT * FROM ORDER_LINE20 UNION ALL
SELECT * FROM ORDER_LINE21 UNION ALL
SELECT * FROM ORDER_LINE22 UNION ALL
SELECT * FROM ORDER_LINE23 UNION ALL
SELECT * FROM ORDER_LINE24 UNION ALL
SELECT * FROM ORDER_LINE25 UNION ALL
SELECT * FROM ORDER_LINE26 UNION ALL
SELECT * FROM ORDER_LINE27 UNION ALL
SELECT * FROM ORDER_LINE28 UNION ALL
SELECT * FROM ORDER_LINE29 UNION ALL
SELECT * FROM ORDER_LINE30 UNION ALL
SELECT * FROM ORDER_LINE31 UNION ALL
SELECT * FROM ORDER_LINE32 UNION ALL
SELECT * FROM ORDER_LINE33 UNION ALL
SELECT * FROM ORDER_LINE34 UNION ALL

```

```

SELECT * FROM ORDER_LINE35 UNION ALL
SELECT * FROM ORDER_LINE36 UNION ALL
SELECT * FROM ORDER_LINE37 UNION ALL
SELECT * FROM ORDER_LINE38
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW STOCK.ddl

```

connect to TPCC in share mode;
DROP VIEW STOCK;
CREATE VIEW STOCK
(S_REMOTE_CNT,
S_QUANTITY,
S_ORDER_CNT,
S_YTD,
S_DATA,
S_DIST_01,
S_DIST_02,
S_DIST_03,
S_DIST_04,
S_DIST_05,
S_DIST_06,
S_DIST_07,
S_DIST_08,
S_DIST_09,
S_DIST_10,
S_I_ID,
S_W_ID
) AS SELECT * FROM STOCK1 UNION ALL
SELECT * FROM STOCK2 UNION ALL
SELECT * FROM STOCK3 UNION ALL
SELECT * FROM STOCK4 UNION ALL
SELECT * FROM STOCK5 UNION ALL
SELECT * FROM STOCK6 UNION ALL
SELECT * FROM STOCK7 UNION ALL
SELECT * FROM STOCK8 UNION ALL
SELECT * FROM STOCK9 UNION ALL
SELECT * FROM STOCK10 UNION ALL
SELECT * FROM STOCK11 UNION ALL
SELECT * FROM STOCK12 UNION ALL
SELECT * FROM STOCK13 UNION ALL
SELECT * FROM STOCK14 UNION ALL
SELECT * FROM STOCK15 UNION ALL
SELECT * FROM STOCK16 UNION ALL
SELECT * FROM STOCK17 UNION ALL
SELECT * FROM STOCK18 UNION ALL
SELECT * FROM STOCK19 UNION ALL
SELECT * FROM STOCK20 UNION ALL
SELECT * FROM STOCK21 UNION ALL
SELECT * FROM STOCK22 UNION ALL
SELECT * FROM STOCK23 UNION ALL
SELECT * FROM STOCK24 UNION ALL
SELECT * FROM STOCK25 UNION ALL
SELECT * FROM STOCK26 UNION ALL
SELECT * FROM STOCK27 UNION ALL
SELECT * FROM STOCK28 UNION ALL
SELECT * FROM STOCK29 UNION ALL
SELECT * FROM STOCK30 UNION ALL
SELECT * FROM STOCK31 UNION ALL
SELECT * FROM STOCK32 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
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

CRVW WAREHOUSE.ddl

```

connect to TPCC in share mode;
DROP VIEW WAREHOUSE;
CREATE VIEW WAREHOUSE
(W_NAME,
W_STREET_1,
W_STREET_2,
W_CITY,
W_STATE,
W_ZIP,
W_TAX,
W_YTD,
W_ID
) AS SELECT * FROM WAREHOUSE1 UNION ALL
SELECT * FROM WAREHOUSE2 UNION ALL
SELECT * FROM WAREHOUSE3 UNION ALL
SELECT * FROM WAREHOUSE4 UNION ALL
SELECT * FROM WAREHOUSE5 UNION ALL
SELECT * FROM WAREHOUSE6 UNION ALL
SELECT * FROM WAREHOUSE7 UNION ALL
SELECT * FROM WAREHOUSE8 UNION ALL
SELECT * FROM WAREHOUSE9 UNION ALL
SELECT * FROM WAREHOUSE10 UNION ALL
SELECT * FROM WAREHOUSE11 UNION ALL
SELECT * FROM WAREHOUSE12 UNION ALL
SELECT * FROM WAREHOUSE13 UNION ALL
SELECT * FROM WAREHOUSE14 UNION ALL
SELECT * FROM WAREHOUSE15 UNION ALL
SELECT * FROM WAREHOUSE16 UNION ALL
SELECT * FROM WAREHOUSE17 UNION ALL
SELECT * FROM WAREHOUSE18 UNION ALL

```

```

SELECT * FROM WAREHOUSE19 UNION ALL
SELECT * FROM WAREHOUSE20 UNION ALL
SELECT * FROM WAREHOUSE21 UNION ALL
SELECT * FROM WAREHOUSE22 UNION ALL
SELECT * FROM WAREHOUSE23 UNION ALL
SELECT * FROM WAREHOUSE24 UNION ALL
SELECT * FROM WAREHOUSE25 UNION ALL
SELECT * FROM WAREHOUSE26 UNION ALL
SELECT * FROM WAREHOUSE27 UNION ALL
SELECT * FROM WAREHOUSE28 UNION ALL
SELECT * FROM WAREHOUSE29 UNION ALL
SELECT * FROM WAREHOUSE30 UNION ALL
SELECT * FROM WAREHOUSE31 UNION ALL
SELECT * FROM WAREHOUSE32 UNION ALL
SELECT * FROM WAREHOUSE33 UNION ALL
SELECT * FROM WAREHOUSE34 UNION ALL
SELECT * FROM WAREHOUSE35 UNION ALL
SELECT * FROM WAREHOUSE36 UNION ALL
SELECT * FROM WAREHOUSE37 UNION ALL
SELECT * FROM WAREHOUSE38
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;

```

GEN_CUSTOMER.sh

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 717 -f1
/flats/ua_01/customer_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 718 1434 -f1
/flats/ua_01/customer_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1435 2150 -f1
/flats/ua_01/customer_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2151 2867 -f1
/flats/ua_02/customer_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2868 3584 -f1
/flats/ua_02/customer_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3585 4300 -f1
/flats/ua_02/customer_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4301 5017 -f1
/flats/ua_03/customer_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5018 5734 -f1
/flats/ua_03/customer_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5735 6450 -f1
/flats/ua_03/customer_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6451 7167 -f1
/flats/ua_04/customer_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7168 7884 -f1
/flats/ua_04/customer_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7885 8600 -f1
/flats/ua_04/customer_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8601 9317 -f1
/flats/ua_05/customer_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9318 10034 -f1
/flats/ua_05/customer_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10035 10750 -f1
/flats/ua_05/customer_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10751 11467 -f1
/flats/ua_06/customer_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11468 12184 -f1
/flats/ua_06/customer_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12185 12900 -f1
/flats/ua_06/customer_06_3.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12901 13617 -f1
/flats/ua_07/customer_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13618 14334 -f1
/flats/ua_07/customer_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14335 15050 -f1
/flats/ua_07/customer_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15051 15767 -f1
/flats/ua_08/customer_08_1.dat

```

```

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15768 16484 -f1
/flats/ua_08/customer_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16485 17200 -f1
/flats/ua_08/customer_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17201 17917 -f1
/flats/ua_09/customer_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17918 18634 -f1
/flats/ua_09/customer_09_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18635 19350 -f1
/flats/ua_09/customer_09_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19351 20067 -f1
/flats/ua_10/customer_10_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20068 20784 -f1
/flats/ua_10/customer_10_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20785 21500 -f1
/flats/ua_10/customer_10_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21501 22217 -f1
/flats/ua_11/customer_11_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22218 22934 -f1
/flats/ua_11/customer_11_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22935 23650 -f1
/flats/ua_11/customer_11_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23651 24367 -f1
/flats/ua_12/customer_12_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24368 25084 -f1
/flats/ua_12/customer_12_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25085 25800 -f1
/flats/ua_12/customer_12_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25801 26517 -f1
/flats/ua_13/customer_13_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26518 27234 -f1
/flats/ua_13/customer_13_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27235 27950 -f1
/flats/ua_13/customer_13_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27951 28667 -f1
/flats/ua_14/customer_14_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28668 29384 -f1
/flats/ua_14/customer_14_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 29385 30100 -f1
/flats/ua_14/customer_14_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30101 30817 -f1
/flats/ua_15/customer_15_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30818 31534 -f1
/flats/ua_15/customer_15_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31535 32250 -f1
/flats/ua_15/customer_15_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32251 32967 -f1
/flats/ua_16/customer_16_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32968 33684 -f1
/flats/ua_16/customer_16_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33685 34400 -f1
/flats/ua_16/customer_16_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34401 35117 -f1
/flats/ua_17/customer_17_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35118 35834 -f1
/flats/ua_17/customer_17_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35835 36550 -f1
/flats/ua_17/customer_17_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36551 37267 -f1
/flats/ua_18/customer_18_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37268 37984 -f1
/flats/ua_18/customer_18_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37985 38700 -f1
/flats/ua_18/customer_18_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38701 39417 -f1
/flats/ua_19/customer_19_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39418 40134 -f1
/flats/ua_19/customer_19_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40135 40850 -f1
/flats/ua_19/customer_19_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40851 41567 -f1
/flats/ua_20/customer_20_1.dat

```


/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 41568 42284 -f1
/flats/ua_20/customer_20_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 42285 43000 -f1
/flats/ua_20/customer_20_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43001 43717 -f1
/flats/ua_21/customer_21_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43718 44434 -f1
/flats/ua_21/customer_21_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 44435 45150 -f1
/flats/ua_21/customer_21_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 45151 45867 -f1
/flats/ua_22/customer_22_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 45868 46584 -f1
/flats/ua_22/customer_22_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 46585 47300 -f1
/flats/ua_22/customer_22_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 47301 48017 -f1
/flats/ua_23/customer_23_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 48018 48734 -f1
/flats/ua_23/customer_23_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 48735 49450 -f1
/flats/ua_23/customer_23_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 49451 50167 -f1
/flats/ua_24/customer_24_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50168 50884 -f1
/flats/ua_24/customer_24_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50885 51600 -f1
/flats/ua_24/customer_24_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 51601 52317 -f1
/flats/ua_25/customer_25_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 52318 53034 -f1
/flats/ua_25/customer_25_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 53035 53750 -f1
/flats/ua_25/customer_25_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 53751 54467 -f1
/flats/ua_26/customer_26_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 54468 55184 -f1
/flats/ua_26/customer_26_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 55185 55900 -f1
/flats/ua_26/customer_26_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 55901 56617 -f1
/flats/ua_27/customer_27_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 56618 57334 -f1
/flats/ua_27/customer_27_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 57335 58050 -f1
/flats/ua_27/customer_27_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 58051 58767 -f1
/flats/ua_28/customer_28_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 58768 59484 -f1
/flats/ua_28/customer_28_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 59485 60200 -f1
/flats/ua_28/customer_28_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 60201 60917 -f1
/flats/ua_29/customer_29_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 60918 61634 -f1
/flats/ua_29/customer_29_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 61635 62350 -f1
/flats/ua_29/customer_29_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 62351 63067 -f1
/flats/ua_30/customer_30_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 63068 63784 -f1
/flats/ua_30/customer_30_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 63785 64500 -f1
/flats/ua_30/customer_30_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 64501 65217 -f1
/flats/ua_31/customer_31_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 65218 65934 -f1
/flats/ua_31/customer_31_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 65935 66650 -f1
/flats/ua_31/customer_31_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 66651 67367 -f1
/flats/ua_32/customer_32_1.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 67368 68084 -f1
/flats/ua_32/customer_32_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 68085 68800 -f1
/flats/ua_32/customer_32_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 68801 69517 -f1
/flats/ua_33/customer_33_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 69518 70234 -f1
/flats/ua_33/customer_33_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 70235 70950 -f1
/flats/ua_33/customer_33_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 70951 71667 -f1
/flats/ua_34/customer_34_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 71668 72384 -f1
/flats/ua_34/customer_34_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 72385 73100 -f1
/flats/ua_34/customer_34_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 73101 73817 -f1
/flats/ua_35/customer_35_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 73818 74534 -f1
/flats/ua_35/customer_35_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 74535 75250 -f1
/flats/ua_35/customer_35_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 75251 75967 -f1
/flats/ua_36/customer_36_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 75968 76684 -f1
/flats/ua_36/customer_36_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 76685 77400 -f1
/flats/ua_36/customer_36_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 77401 78117 -f1
/flats/ua_37/customer_37_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 78118 78834 -f1
/flats/ua_37/customer_37_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 78835 79550 -f1
/flats/ua_37/customer_37_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 79551 80267 -f1
/flats/ua_38/customer_38_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 80268 80984 -f1
/flats/ua_38/customer_38_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 80985 81700 -f1
/flats/ua_38/customer_38_3.dat

GEN DISTRICT.sh

/home/tpc-c.ibm/dbgen/gendata -t 4 -r 1 2150 -f1 /flats/ua_01/district_01_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 2151 4300 -f1
/flats/ua_02/district_02_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 4301 6450 -f1
/flats/ua_03/district_03_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 6451 8600 -f1
/flats/ua_04/district_04_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 8601 10750 -f1
/flats/ua_05/district_05_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 10751 12900 -f1
/flats/ua_06/district_06_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 12901 15050 -f1
/flats/ua_07/district_07_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 15051 17200 -f1
/flats/ua_08/district_08_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 17201 19350 -f1
/flats/ua_09/district_09_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 19351 21500 -f1
/flats/ua_10/district_10_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 21501 23650 -f1
/flats/ua_11/district_11_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 23651 25800 -f1
/flats/ua_12/district_12_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 25801 27950 -f1
/flats/ua_13/district_13_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 27951 30100 -f1
/flats/ua_14/district_14_1.dat

/home/tpc-c.ibm/dbgen/gendata -t 4 -r 30101 32250 -f1
/flats/ua_15/district_15_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 32251 34400 -f1
/flats/ua_16/district_16_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 34401 36550 -f1
/flats/ua_17/district_17_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 36551 38700 -f1
/flats/ua_18/district_18_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 38701 40850 -f1
/flats/ua_19/district_19_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 40851 43000 -f1
/flats/ua_20/district_20_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 43001 45150 -f1
/flats/ua_21/district_21_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 45151 47300 -f1
/flats/ua_22/district_22_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 47301 49450 -f1
/flats/ua_23/district_23_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 49451 51600 -f1
/flats/ua_24/district_24_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 51601 53750 -f1
/flats/ua_25/district_25_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 53751 55900 -f1
/flats/ua_26/district_26_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 55901 58050 -f1
/flats/ua_27/district_27_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 58051 60200 -f1
/flats/ua_28/district_28_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 60201 62350 -f1
/flats/ua_29/district_29_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 62351 64500 -f1
/flats/ua_30/district_30_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 64501 66650 -f1
/flats/ua_31/district_31_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 66651 68800 -f1
/flats/ua_32/district_32_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 68801 70950 -f1
/flats/ua_33/district_33_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 70951 73100 -f1
/flats/ua_34/district_34_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 73101 75250 -f1
/flats/ua_35/district_35_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 75251 77400 -f1
/flats/ua_36/district_36_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 77401 79550 -f1
/flats/ua_37/district_37_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 4 -r 79551 81700 -f1
/flats/ua_38/district_38_1.dat

GEN HISTORY.sh

/home/tpc-c.ibm/dbgen/gendata -t 8 -r 1 2150 -f1 /flats/ua_01/history_01_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 2151 4300 -f1
/flats/ua_02/history_02_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 4301 6450 -f1
/flats/ua_03/history_03_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 6451 8600 -f1
/flats/ua_04/history_04_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 8601 10750 -f1
/flats/ua_05/history_05_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 10751 12900 -f1
/flats/ua_06/history_06_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 12901 15050 -f1
/flats/ua_07/history_07_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 15051 17200 -f1
/flats/ua_08/history_08_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 17201 19350 -f1
/flats/ua_09/history_09_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 19351 21500 -f1
/flats/ua_10/history_10_1.dat

/home/tpc-c.ibm/dbgen/gendata -t 8 -r 21501 23650 -f1
/flats/ua_11/history_11_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 23651 25800 -f1
/flats/ua_12/history_12_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 25801 27950 -f1
/flats/ua_13/history_13_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 27951 30100 -f1
/flats/ua_14/history_14_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 30101 32250 -f1
/flats/ua_15/history_15_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 32251 34400 -f1
/flats/ua_16/history_16_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 34401 36550 -f1
/flats/ua_17/history_17_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 36551 38700 -f1
/flats/ua_18/history_18_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 38701 40850 -f1
/flats/ua_19/history_19_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 40851 43000 -f1
/flats/ua_20/history_20_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 43001 45150 -f1
/flats/ua_21/history_21_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 45151 47300 -f1
/flats/ua_22/history_22_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 47301 49450 -f1
/flats/ua_23/history_23_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 49451 51600 -f1
/flats/ua_24/history_24_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 51601 53750 -f1
/flats/ua_25/history_25_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 53751 55900 -f1
/flats/ua_26/history_26_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 55901 58050 -f1
/flats/ua_27/history_27_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 58051 60200 -f1
/flats/ua_28/history_28_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 60201 62350 -f1
/flats/ua_29/history_29_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 62351 64500 -f1
/flats/ua_30/history_30_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 64501 66650 -f1
/flats/ua_31/history_31_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 66651 68800 -f1
/flats/ua_32/history_32_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 68801 70950 -f1
/flats/ua_33/history_33_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 70951 73100 -f1
/flats/ua_34/history_34_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 73101 75250 -f1
/flats/ua_35/history_35_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 75251 77400 -f1
/flats/ua_36/history_36_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 77401 79550 -f1
/flats/ua_37/history_37_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 8 -r 79551 81700 -f1
/flats/ua_38/history_38_1.dat

GEN ITEM.sh

/home/tpc-c.ibm/dbgen/gendata -t 5 -f1 /flats/ua_00/item_1.dat

GEN NEW ORDER.sh

/home/tpc-c.ibm/dbgen/gendata -t 11 -r 1 2150 -f1
/flats/ua_01/neworder_01_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 2151 4300 -f1
/flats/ua_02/neworder_02_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 4301 6450 -f1
/flats/ua_03/neworder_03_1.dat

/home/tpc-c.ibm/dbgen/gendata -t 11 -r 6451 8600 -f1
/flats/ua_04/neworder_04_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 8601 10750 -f1
/flats/ua_05/neworder_05_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 10751 12900 -f1
/flats/ua_06/neworder_06_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 12901 15050 -f1
/flats/ua_07/neworder_07_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 15051 17200 -f1
/flats/ua_08/neworder_08_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 17201 19350 -f1
/flats/ua_09/neworder_09_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 19351 21500 -f1
/flats/ua_10/neworder_10_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 21501 23650 -f1
/flats/ua_11/neworder_11_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 23651 25800 -f1
/flats/ua_12/neworder_12_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 25801 27950 -f1
/flats/ua_13/neworder_13_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 27951 30100 -f1
/flats/ua_14/neworder_14_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 30101 32250 -f1
/flats/ua_15/neworder_15_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 32251 34400 -f1
/flats/ua_16/neworder_16_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 34401 36550 -f1
/flats/ua_17/neworder_17_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 36551 38700 -f1
/flats/ua_18/neworder_18_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 38701 40850 -f1
/flats/ua_19/neworder_19_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 40851 43000 -f1
/flats/ua_20/neworder_20_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 43001 45150 -f1
/flats/ua_21/neworder_21_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 45151 47300 -f1
/flats/ua_22/neworder_22_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 47301 49450 -f1
/flats/ua_23/neworder_23_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 49451 51600 -f1
/flats/ua_24/neworder_24_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 51601 53750 -f1
/flats/ua_25/neworder_25_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 53751 55900 -f1
/flats/ua_26/neworder_26_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 55901 58050 -f1
/flats/ua_27/neworder_27_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 58051 60200 -f1
/flats/ua_28/neworder_28_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 60201 62350 -f1
/flats/ua_29/neworder_29_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 62351 64500 -f1
/flats/ua_30/neworder_30_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 64501 66650 -f1
/flats/ua_31/neworder_31_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 66651 68800 -f1
/flats/ua_32/neworder_32_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 68801 70950 -f1
/flats/ua_33/neworder_33_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 70951 73100 -f1
/flats/ua_34/neworder_34_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 73101 75250 -f1
/flats/ua_35/neworder_35_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 75251 77400 -f1
/flats/ua_36/neworder_36_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 77401 79550 -f1
/flats/ua_37/neworder_37_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 11 -r 79551 81700 -f1
/flats/ua_38/neworder_38_1.dat

GEN ORDERS.sh

/home/tpc-c.ibm/dbgen/gendata -t 9 -r 1 1075 -f1
/flats/ua_00_01/orders_01_1.dat -f2 /flats/ua_01/orderline_01_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 1076 2150 -f1
/flats/ua_00_01/orders_01_2.dat -f2 /flats/ua_01/orderline_01_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 2151 3225 -f1
/flats/ua_00_02/orders_02_1.dat -f2 /flats/ua_02/orderline_02_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 3226 4300 -f1
/flats/ua_00_02/orders_02_2.dat -f2 /flats/ua_02/orderline_02_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 4301 5375 -f1
/flats/ua_00_03/orders_03_1.dat -f2 /flats/ua_03/orderline_03_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 5376 6450 -f1
/flats/ua_00_03/orders_03_2.dat -f2 /flats/ua_03/orderline_03_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 6451 7525 -f1
/flats/ua_00_04/orders_04_1.dat -f2 /flats/ua_04/orderline_04_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 7526 8600 -f1
/flats/ua_00_04/orders_04_2.dat -f2 /flats/ua_04/orderline_04_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 8601 9675 -f1
/flats/ua_00_05/orders_05_1.dat -f2 /flats/ua_05/orderline_05_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 9676 10750 -f1
/flats/ua_00_05/orders_05_2.dat -f2 /flats/ua_05/orderline_05_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 10751 11825 -f1
/flats/ua_00_06/orders_06_1.dat -f2 /flats/ua_06/orderline_06_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 11826 12900 -f1
/flats/ua_00_06/orders_06_2.dat -f2 /flats/ua_06/orderline_06_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 12901 13975 -f1
/flats/ua_00_07/orders_07_1.dat -f2 /flats/ua_07/orderline_07_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 13976 15050 -f1
/flats/ua_00_07/orders_07_2.dat -f2 /flats/ua_07/orderline_07_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 15051 16125 -f1
/flats/ua_00_08/orders_08_1.dat -f2 /flats/ua_08/orderline_08_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 16126 17200 -f1
/flats/ua_00_08/orders_08_2.dat -f2 /flats/ua_08/orderline_08_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 17201 18275 -f1
/flats/ua_00_09/orders_09_1.dat -f2 /flats/ua_09/orderline_09_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 18276 19350 -f1
/flats/ua_00_09/orders_09_2.dat -f2 /flats/ua_09/orderline_09_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 19351 20425 -f1
/flats/ua_00_10/orders_10_1.dat -f2 /flats/ua_10/orderline_10_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 20426 21500 -f1
/flats/ua_00_10/orders_10_2.dat -f2 /flats/ua_10/orderline_10_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 21501 22575 -f1
/flats/ua_00_11/orders_11_1.dat -f2 /flats/ua_11/orderline_11_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 22576 23650 -f1
/flats/ua_00_11/orders_11_2.dat -f2 /flats/ua_11/orderline_11_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 23651 24725 -f1
/flats/ua_00_12/orders_12_1.dat -f2 /flats/ua_12/orderline_12_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 24726 25800 -f1
/flats/ua_00_12/orders_12_2.dat -f2 /flats/ua_12/orderline_12_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 25801 26875 -f1
/flats/ua_00_13/orders_13_1.dat -f2 /flats/ua_13/orderline_13_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 26876 27950 -f1
/flats/ua_00_13/orders_13_2.dat -f2 /flats/ua_13/orderline_13_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 27951 29025 -f1
/flats/ua_00_14/orders_14_1.dat -f2 /flats/ua_14/orderline_14_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 29026 30100 -f1
/flats/ua_00_14/orders_14_2.dat -f2 /flats/ua_14/orderline_14_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 30101 31175 -f1
/flats/ua_00_15/orders_15_1.dat -f2 /flats/ua_15/orderline_15_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 31176 32250 -f1
/flats/ua_00_15/orders_15_2.dat -f2 /flats/ua_15/orderline_15_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 32251 33325 -f1
/flats/ua_00_16/orders_16_1.dat -f2 /flats/ua_16/orderline_16_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 33326 34400 -f1
/flats/ua_00_16/orders_16_2.dat -f2 /flats/ua_16/orderline_16_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 34401 35475 -f1
/flats/ua_00_17/orders_17_1.dat -f2 /flats/ua_17/orderline_17_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 35476 36550 -f1
/flats/ua_00_17/orders_17_2.dat -f2 /flats/ua_17/orderline_17_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 36551 37625 -f1
/flats/ua_00_18/orders_18_1.dat -f2 /flats/ua_18/orderline_18_1.dat

/home/tpc-c.ibm/dbgen/gendata -t 9 -r 37626 38700 -f1
/flats/ua_00_18/orders_18_2.dat -f2 /flats/ua_18/orderline_18_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 38701 39775 -f1
/flats/ua_00_19/orders_19_1.dat -f2 /flats/ua_19/orderline_19_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 39776 40850 -f1
/flats/ua_00_19/orders_19_2.dat -f2 /flats/ua_19/orderline_19_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 40851 41925 -f1
/flats/ua_00_20/orders_20_1.dat -f2 /flats/ua_20/orderline_20_1.dat

/home/tpc-c.ibm/dbgen/gendata -t 9 -r 41926 43000 -f1
/flats/ua_00_20/orders_20_2.dat -f2 /flats/ua_20/orderline_20_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 43001 44075 -f1
/flats/ua_00_21/orders_21_1.dat -f2 /flats/ua_21/orderline_21_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 44076 45150 -f1
/flats/ua_00_21/orders_21_2.dat -f2 /flats/ua_21/orderline_21_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 45151 46225 -f1
/flats/ua_00_22/orders_22_1.dat -f2 /flats/ua_22/orderline_22_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 46226 47300 -f1
/flats/ua_00_22/orders_22_2.dat -f2 /flats/ua_22/orderline_22_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 47301 48375 -f1
/flats/ua_00_23/orders_23_1.dat -f2 /flats/ua_23/orderline_23_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 48376 49450 -f1
/flats/ua_00_23/orders_23_2.dat -f2 /flats/ua_23/orderline_23_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 49451 50525 -f1
/flats/ua_00_24/orders_24_1.dat -f2 /flats/ua_24/orderline_24_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 50526 51600 -f1
/flats/ua_00_24/orders_24_2.dat -f2 /flats/ua_24/orderline_24_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 51601 52675 -f1
/flats/ua_00_25/orders_25_1.dat -f2 /flats/ua_25/orderline_25_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 52676 53750 -f1
/flats/ua_00_25/orders_25_2.dat -f2 /flats/ua_25/orderline_25_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 53751 54825 -f1
/flats/ua_00_26/orders_26_1.dat -f2 /flats/ua_26/orderline_26_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 54826 55900 -f1
/flats/ua_00_26/orders_26_2.dat -f2 /flats/ua_26/orderline_26_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 55901 56975 -f1
/flats/ua_00_27/orders_27_1.dat -f2 /flats/ua_27/orderline_27_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 56976 58050 -f1
/flats/ua_00_27/orders_27_2.dat -f2 /flats/ua_27/orderline_27_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 58051 59125 -f1
/flats/ua_00_28/orders_28_1.dat -f2 /flats/ua_28/orderline_28_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 59126 60200 -f1
/flats/ua_00_28/orders_28_2.dat -f2 /flats/ua_28/orderline_28_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 60201 61275 -f1
/flats/ua_00_29/orders_29_1.dat -f2 /flats/ua_29/orderline_29_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 61276 62350 -f1
/flats/ua_00_29/orders_29_2.dat -f2 /flats/ua_29/orderline_29_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 62351 63425 -f1
/flats/ua_00_30/orders_30_1.dat -f2 /flats/ua_30/orderline_30_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 63426 64500 -f1
/flats/ua_00_30/orders_30_2.dat -f2 /flats/ua_30/orderline_30_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 64501 65575 -f1
/flats/ua_00_31/orders_31_1.dat -f2 /flats/ua_31/orderline_31_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 65576 66650 -f1
/flats/ua_00_31/orders_31_2.dat -f2 /flats/ua_31/orderline_31_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 66651 67725 -f1
/flats/ua_00_32/orders_32_1.dat -f2 /flats/ua_32/orderline_32_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 67726 68800 -f1
/flats/ua_00_32/orders_32_2.dat -f2 /flats/ua_32/orderline_32_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 68801 69875 -f1
/flats/ua_00_33/orders_33_1.dat -f2 /flats/ua_33/orderline_33_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 69876 70950 -f1
/flats/ua_00_33/orders_33_2.dat -f2 /flats/ua_33/orderline_33_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 70951 72025 -f1
/flats/ua_00_34/orders_34_1.dat -f2 /flats/ua_34/orderline_34_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 72026 73100 -f1
/flats/ua_00_34/orders_34_2.dat -f2 /flats/ua_34/orderline_34_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 73101 74175 -f1
/flats/ua_00_35/orders_35_1.dat -f2 /flats/ua_35/orderline_35_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 74176 75250 -f1
/flats/ua_00_35/orders_35_2.dat -f2 /flats/ua_35/orderline_35_2.dat

/home/tpc-c.ibm/dbgen/gendata -t 9 -r 75251 76325 -f1
/flats/ua_00_36/orders_36_1.dat -f2 /flats/ua_36/orderline_36_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 76326 77400 -f1
/flats/ua_00_36/orders_36_2.dat -f2 /flats/ua_36/orderline_36_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 77401 78475 -f1
/flats/ua_00_37/orders_37_1.dat -f2 /flats/ua_37/orderline_37_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 78476 79550 -f1
/flats/ua_00_37/orders_37_2.dat -f2 /flats/ua_37/orderline_37_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 79551 80625 -f1
/flats/ua_00_38/orders_38_1.dat -f2 /flats/ua_38/orderline_38_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 9 -r 80626 81700 -f1
/flats/ua_00_38/orders_38_2.dat -f2 /flats/ua_38/orderline_38_2.dat

GEN STOCK.sh

/home/tpc-c.ibm/dbgen/gendata -t 6 -r 1 215 -f1 /flats/ua_01/stock_01_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 216 430 -f1 /flats/ua_01/stock_01_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 431 645 -f1 /flats/ua_01/stock_01_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 646 860 -f1 /flats/ua_01/stock_01_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 861 1075 -f1 /flats/ua_01/stock_01_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 1076 1290 -f1
/flats/ua_02/stock_02_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 1291 1505 -f1
/flats/ua_02/stock_02_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 1506 1720 -f1
/flats/ua_02/stock_02_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 1721 1935 -f1
/flats/ua_02/stock_02_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 1936 2150 -f1
/flats/ua_02/stock_02_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 2151 2365 -f1
/flats/ua_03/stock_03_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 2366 2580 -f1
/flats/ua_03/stock_03_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 2581 2795 -f1
/flats/ua_03/stock_03_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 2796 3010 -f1
/flats/ua_03/stock_03_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 3011 3225 -f1
/flats/ua_03/stock_03_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 3226 3440 -f1
/flats/ua_04/stock_04_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 3441 3655 -f1
/flats/ua_04/stock_04_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 3656 3870 -f1
/flats/ua_04/stock_04_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 3871 4085 -f1
/flats/ua_04/stock_04_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 4086 4300 -f1
/flats/ua_04/stock_04_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 4301 4515 -f1
/flats/ua_05/stock_05_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 4516 4730 -f1
/flats/ua_05/stock_05_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 4731 4945 -f1
/flats/ua_05/stock_05_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 4946 5160 -f1
/flats/ua_05/stock_05_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 5161 5375 -f1
/flats/ua_05/stock_05_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 5376 5590 -f1
/flats/ua_06/stock_06_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 5591 5805 -f1
/flats/ua_06/stock_06_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 5806 6020 -f1
/flats/ua_06/stock_06_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 6021 6235 -f1
/flats/ua_06/stock_06_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 6236 6450 -f1
/flats/ua_06/stock_06_5.dat

/home/tpc-c.ibm/dbgen/gendata -t 6 -r 6451 6665 -f1
/flats/ua_07/stock_07_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 6666 6880 -f1
/flats/ua_07/stock_07_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 6881 7095 -f1
/flats/ua_07/stock_07_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 7096 7310 -f1
/flats/ua_07/stock_07_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 7311 7525 -f1
/flats/ua_07/stock_07_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 7526 7740 -f1
/flats/ua_08/stock_08_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 7741 7955 -f1
/flats/ua_08/stock_08_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 7956 8170 -f1
/flats/ua_08/stock_08_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 8171 8385 -f1
/flats/ua_08/stock_08_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 8386 8600 -f1
/flats/ua_08/stock_08_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 8601 8815 -f1
/flats/ua_09/stock_09_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 8816 9030 -f1
/flats/ua_09/stock_09_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 9031 9245 -f1
/flats/ua_09/stock_09_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 9246 9460 -f1
/flats/ua_09/stock_09_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 9461 9675 -f1
/flats/ua_09/stock_09_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 9676 9890 -f1
/flats/ua_10/stock_10_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 9891 10105 -f1
/flats/ua_10/stock_10_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 10106 10320 -f1
/flats/ua_10/stock_10_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 10321 10535 -f1
/flats/ua_10/stock_10_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 10536 10750 -f1
/flats/ua_10/stock_10_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 10751 10965 -f1
/flats/ua_11/stock_11_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 10966 11180 -f1
/flats/ua_11/stock_11_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 11181 11395 -f1
/flats/ua_11/stock_11_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 11396 11610 -f1
/flats/ua_11/stock_11_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 11611 11825 -f1
/flats/ua_11/stock_11_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 11826 12040 -f1
/flats/ua_12/stock_12_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 12041 12255 -f1
/flats/ua_12/stock_12_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 12256 12470 -f1
/flats/ua_12/stock_12_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 12471 12685 -f1
/flats/ua_12/stock_12_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 12686 12900 -f1
/flats/ua_12/stock_12_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 12901 13115 -f1
/flats/ua_13/stock_13_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 13116 13330 -f1
/flats/ua_13/stock_13_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 13331 13545 -f1
/flats/ua_13/stock_13_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 13546 13760 -f1
/flats/ua_13/stock_13_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 13761 13975 -f1
/flats/ua_13/stock_13_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 13976 14190 -f1
/flats/ua_14/stock_14_1.dat

/home/tpc-c.ibm/dbgen/gendata -t 6 -r 14191 14405 -f1
/flats/ua_14/stock_14_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 14406 14620 -f1
/flats/ua_14/stock_14_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 14621 14835 -f1
/flats/ua_14/stock_14_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 14836 15050 -f1
/flats/ua_14/stock_14_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 15051 15265 -f1
/flats/ua_15/stock_15_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 15266 15480 -f1
/flats/ua_15/stock_15_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 15481 15695 -f1
/flats/ua_15/stock_15_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 15696 15910 -f1
/flats/ua_15/stock_15_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 15911 16125 -f1
/flats/ua_15/stock_15_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 16126 16340 -f1
/flats/ua_16/stock_16_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 16341 16555 -f1
/flats/ua_16/stock_16_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 16556 16770 -f1
/flats/ua_16/stock_16_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 16771 16985 -f1
/flats/ua_16/stock_16_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 16986 17200 -f1
/flats/ua_16/stock_16_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 17201 17415 -f1
/flats/ua_17/stock_17_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 17416 17630 -f1
/flats/ua_17/stock_17_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 17631 17845 -f1
/flats/ua_17/stock_17_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 17846 18060 -f1
/flats/ua_17/stock_17_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 18061 18275 -f1
/flats/ua_17/stock_17_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 18276 18490 -f1
/flats/ua_18/stock_18_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 18491 18705 -f1
/flats/ua_18/stock_18_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 18706 18920 -f1
/flats/ua_18/stock_18_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 18921 19135 -f1
/flats/ua_18/stock_18_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 19136 19350 -f1
/flats/ua_18/stock_18_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 19351 19565 -f1
/flats/ua_19/stock_19_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 19566 19780 -f1
/flats/ua_19/stock_19_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 19781 19995 -f1
/flats/ua_19/stock_19_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 19996 20210 -f1
/flats/ua_19/stock_19_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 20211 20425 -f1
/flats/ua_19/stock_19_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 20426 20640 -f1
/flats/ua_20/stock_20_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 20641 20855 -f1
/flats/ua_20/stock_20_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 20856 21070 -f1
/flats/ua_20/stock_20_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 21071 21285 -f1
/flats/ua_20/stock_20_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 21286 21500 -f1
/flats/ua_20/stock_20_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 21501 21715 -f1
/flats/ua_21/stock_21_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 21716 21930 -f1
/flats/ua_21/stock_21_2.dat

/home/tpc-c.ibm/dbgen/gendata -t 6 -r 21931 22145 -f1
/flats/ua_21/stock_21_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 22146 22360 -f1
/flats/ua_21/stock_21_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 22361 22575 -f1
/flats/ua_21/stock_21_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 22576 22790 -f1
/flats/ua_22/stock_22_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 22791 23005 -f1
/flats/ua_22/stock_22_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 23006 23220 -f1
/flats/ua_22/stock_22_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 23221 23435 -f1
/flats/ua_22/stock_22_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 23436 23650 -f1
/flats/ua_22/stock_22_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 23651 23865 -f1
/flats/ua_23/stock_23_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 23866 24080 -f1
/flats/ua_23/stock_23_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 24081 24295 -f1
/flats/ua_23/stock_23_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 24296 24510 -f1
/flats/ua_23/stock_23_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 24511 24725 -f1
/flats/ua_23/stock_23_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 24726 24940 -f1
/flats/ua_24/stock_24_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 24941 25155 -f1
/flats/ua_24/stock_24_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 25156 25370 -f1
/flats/ua_24/stock_24_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 25371 25585 -f1
/flats/ua_24/stock_24_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 25586 25800 -f1
/flats/ua_24/stock_24_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 25801 26015 -f1
/flats/ua_25/stock_25_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 26016 26230 -f1
/flats/ua_25/stock_25_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 26231 26445 -f1
/flats/ua_25/stock_25_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 26446 26660 -f1
/flats/ua_25/stock_25_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 26661 26875 -f1
/flats/ua_25/stock_25_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 26876 27090 -f1
/flats/ua_26/stock_26_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 27091 27305 -f1
/flats/ua_26/stock_26_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 27306 27520 -f1
/flats/ua_26/stock_26_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 27521 27735 -f1
/flats/ua_26/stock_26_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 27736 27950 -f1
/flats/ua_26/stock_26_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 27951 28165 -f1
/flats/ua_27/stock_27_1.dat

/home/tpc-c.ibm/dbgen/gendata -t 6 -r 28166 28380 -f1
/flats/ua_27/stock_27_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 28381 28595 -f1
/flats/ua_27/stock_27_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 28596 28810 -f1
/flats/ua_27/stock_27_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 28811 29025 -f1
/flats/ua_27/stock_27_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 29026 29240 -f1
/flats/ua_28/stock_28_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 29241 29455 -f1
/flats/ua_28/stock_28_2.dat

/home/tpc-c.ibm/dbgen/gendata -t 6 -r 29456 29670 -f1
/flats/ua_28/stock_28_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 29671 29885 -f1
/flats/ua_28/stock_28_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 29886 30100 -f1
/flats/ua_28/stock_28_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 30101 30315 -f1
/flats/ua_29/stock_29_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 30316 30530 -f1
/flats/ua_29/stock_29_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 30531 30745 -f1
/flats/ua_29/stock_29_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 30746 30960 -f1
/flats/ua_29/stock_29_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 30961 31175 -f1
/flats/ua_29/stock_29_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 31176 31390 -f1
/flats/ua_30/stock_30_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 31391 31605 -f1
/flats/ua_30/stock_30_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 31606 31820 -f1
/flats/ua_30/stock_30_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 31821 32035 -f1
/flats/ua_30/stock_30_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 32036 32250 -f1
/flats/ua_30/stock_30_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 32251 32465 -f1
/flats/ua_31/stock_31_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 32466 32680 -f1
/flats/ua_31/stock_31_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 32681 32895 -f1
/flats/ua_31/stock_31_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 32896 33110 -f1
/flats/ua_31/stock_31_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 33111 33325 -f1
/flats/ua_31/stock_31_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 33326 33540 -f1
/flats/ua_32/stock_32_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 33541 33755 -f1
/flats/ua_32/stock_32_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 33756 33970 -f1
/flats/ua_32/stock_32_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 33971 34185 -f1
/flats/ua_32/stock_32_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 34186 34400 -f1
/flats/ua_32/stock_32_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 34401 34615 -f1
/flats/ua_33/stock_33_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 34616 34830 -f1
/flats/ua_33/stock_33_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 34831 35045 -f1
/flats/ua_33/stock_33_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 35046 35260 -f1
/flats/ua_33/stock_33_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 35261 35475 -f1
/flats/ua_33/stock_33_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 35476 35690 -f1
/flats/ua_34/stock_34_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 35691 35905 -f1
/flats/ua_34/stock_34_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 35906 36120 -f1
/flats/ua_34/stock_34_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 36121 36335 -f1
/flats/ua_34/stock_34_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 36336 36550 -f1
/flats/ua_34/stock_34_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 36551 36765 -f1
/flats/ua_35/stock_35_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 36766 36980 -f1
/flats/ua_35/stock_35_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 36981 37195 -f1
/flats/ua_35/stock_35_3.dat

```

/home/tpc-c.ibm/dbgen/gendata -t 6 -r 37196 37410 -f1
/flats/ua_35/stock_35_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 37411 37625 -f1
/flats/ua_35/stock_35_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 37626 37840 -f1
/flats/ua_36/stock_36_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 37841 38055 -f1
/flats/ua_36/stock_36_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 38056 38270 -f1
/flats/ua_36/stock_36_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 38271 38485 -f1
/flats/ua_36/stock_36_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 38486 38700 -f1
/flats/ua_36/stock_36_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 38701 38915 -f1
/flats/ua_37/stock_37_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 38916 39130 -f1
/flats/ua_37/stock_37_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 39131 39345 -f1
/flats/ua_37/stock_37_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 39346 39560 -f1
/flats/ua_37/stock_37_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 39561 39775 -f1
/flats/ua_37/stock_37_5.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 39776 39990 -f1
/flats/ua_38/stock_38_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 39991 40205 -f1
/flats/ua_38/stock_38_2.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 40206 40420 -f1
/flats/ua_38/stock_38_3.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 40421 40635 -f1
/flats/ua_38/stock_38_4.dat
/home/tpc-c.ibm/dbgen/gendata -t 6 -r 40636 40850 -f1
/flats/ua_38/stock_38_5.dat

```

GEN WAREHOUSE.sh

```

/home/tpc-c.ibm/dbgen/gendata -t 3 -r 1 2150 -f1
/flats/ua_01/warehouse_01_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 2151 4300 -f1
/flats/ua_02/warehouse_02_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 4301 6450 -f1
/flats/ua_03/warehouse_03_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 6451 8600 -f1
/flats/ua_04/warehouse_04_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 8601 10750 -f1
/flats/ua_05/warehouse_05_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 10751 12900 -f1
/flats/ua_06/warehouse_06_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 12901 15050 -f1
/flats/ua_07/warehouse_07_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 15051 17200 -f1
/flats/ua_08/warehouse_08_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 17201 19350 -f1
/flats/ua_09/warehouse_09_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 19351 21500 -f1
/flats/ua_10/warehouse_10_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 21501 23650 -f1
/flats/ua_11/warehouse_11_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 23651 25800 -f1
/flats/ua_12/warehouse_12_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 25801 27950 -f1
/flats/ua_13/warehouse_13_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 27951 30100 -f1
/flats/ua_14/warehouse_14_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 30101 32250 -f1
/flats/ua_15/warehouse_15_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 32251 34400 -f1
/flats/ua_16/warehouse_16_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 34401 36550 -f1
/flats/ua_17/warehouse_17_1.dat

```

```

/home/tpc-c.ibm/dbgen/gendata -t 3 -r 36551 38700 -f1
/flats/ua_18/warehouse_18_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 38701 40850 -f1
/flats/ua_19/warehouse_19_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 40851 43000 -f1
/flats/ua_20/warehouse_20_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 43001 45150 -f1
/flats/ua_21/warehouse_21_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 45151 47300 -f1
/flats/ua_22/warehouse_22_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 47301 49450 -f1
/flats/ua_23/warehouse_23_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 49451 51600 -f1
/flats/ua_24/warehouse_24_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 51601 53750 -f1
/flats/ua_25/warehouse_25_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 53751 55900 -f1
/flats/ua_26/warehouse_26_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 55901 58050 -f1
/flats/ua_27/warehouse_27_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 58051 60200 -f1
/flats/ua_28/warehouse_28_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 60201 62350 -f1
/flats/ua_29/warehouse_29_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 62351 64500 -f1
/flats/ua_30/warehouse_30_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 64501 66650 -f1
/flats/ua_31/warehouse_31_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 66651 68800 -f1
/flats/ua_32/warehouse_32_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 68801 70950 -f1
/flats/ua_33/warehouse_33_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 70951 73100 -f1
/flats/ua_34/warehouse_34_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 73101 75250 -f1
/flats/ua_35/warehouse_35_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 75251 77400 -f1
/flats/ua_36/warehouse_36_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 77401 79550 -f1
/flats/ua_37/warehouse_37_1.dat
/home/tpc-c.ibm/dbgen/gendata -t 3 -r 79551 81700 -f1
/flats/ua_38/warehouse_38_1.dat

```

LOAD CUSTOMER ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/customer_01_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/customer_01_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/customer_01_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;

```



```

COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/customer_38_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/customer_38_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/customer_38_3.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;

```

LOAD DISTRICT ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_01/district_01_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/ua_02/district_02_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/ua_03/district_03_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/ua_04/district_04_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/ua_05/district_05_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/ua_06/district_06_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/ua_07/district_07_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/ua_08/district_08_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_09/district_09_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_10/district_10_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_11/district_11_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_12/district_12_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_13/district_13_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_14/district_14_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_15/district_15_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_16/district_16_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_17/district_17_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_18/district_18_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_19/district_19_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT19;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_20/district_20_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_21/district_21_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_22/district_22_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_23/district_23_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_24/district_24_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_25/district_25_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_26/district_26_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_27/district_27_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_28/district_28_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_29/district_29_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_30/district_30_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_31/district_31_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT31;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_32/district_32_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_33/district_33_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_34/district_34_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_35/district_35_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_36/district_36_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_37/district_37_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_38/district_38_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO DISTRICT38;
COMMIT WORK;
CONNECT RESET;

```

LOAD_HISTORY_ALL.ddl

```

connect to TPCC in share mode;
LOAD FROM /flats/ua_01/history_01_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY1
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_02/history_02_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY2
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_03/history_03_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY3
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_04/history_04_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY4
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_05/history_05_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY5
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;

```



```

LOAD FROM /flats/ua_34/history_34_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY34
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_35/history_35_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY35
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_36/history_36_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY36
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_37/history_37_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY37
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;
connect to TPCC in share mode;
LOAD FROM /flats/ua_38/history_38_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY38
NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 2 ;
connect reset;

```

LOAD ITEM 1.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_00/item_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 1000 INSERT INTO
ITEM;
COMMIT WORK;
CONNECT RESET;

```

LOAD NEW ORDER ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/neworder_01_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/neworder_02_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/neworder_03_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/neworder_04_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;

```

```

ALTER TABLE NEW_ORDER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/neworder_05_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/neworder_06_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/neworder_07_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/neworder_08_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/neworder_09_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/neworder_10_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/neworder_11_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/neworder_12_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/neworder_13_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;

```



```

UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/neworder_32_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/neworder_33_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/neworder_34_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/neworder_35_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/neworder_36_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/neworder_37_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER37;
COMMIT WORK;
CONNECT RESET;

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE NEW_ORDER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/neworder_38_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO NEW_ORDER38;
COMMIT WORK;
CONNECT RESET;

```

LOAD ORDERS ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_01/orders_01_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS1 ACTIVATE NOT LOGGED INITIALLY;

```

```

IMPORT FROM /flats/ua_00_01/orders_01_2.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_02/orders_02_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;

```

```

ALTER TABLE ORDERS2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_02/orders_02_2.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_03/orders_03_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_03/orders_03_2.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS3;

```

```

COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_04/orders_04_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_04/orders_04_2.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;

```

```

ALTER TABLE ORDERS5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_05/orders_05_1.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDERS5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_00_05/orders_05_2.dat OF DEL MODIFIED BY
COLDEL|KEEPBLANKS COMPOUND=50 COMMITCOUNT 21510000
INSERT INTO ORDERS5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;

```



```

ALTER TABLE ORDER_LINE30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/orderline_30_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/orderline_31_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/orderline_31_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/orderline_32_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/orderline_32_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/orderline_33_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/orderline_33_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/orderline_34_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/orderline_34_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;

```

```

ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/orderline_35_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/orderline_35_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/orderline_36_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/orderline_36_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/orderline_37_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/orderline_37_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/orderline_38_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
UPDATE COMMAND OPTIONS USING C OFF;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/orderline_38_2.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 COMMITCOUNT 330000000
INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;

```

LOAD STOCK_ALL.ddl

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_01/stock_01_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;

```



```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_26/warehouse_26_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_27/warehouse_27_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_28/warehouse_28_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_29/warehouse_29_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_30/warehouse_30_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_31/warehouse_31_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_32/warehouse_32_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_33/warehouse_33_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_34/warehouse_34_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_35/warehouse_35_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_36/warehouse_36_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_37/warehouse_37_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE37;
COMMIT WORK;
CONNECT RESET;

```

```

CONNECT TO TPCC IN SHARE MODE;
IMPORT FROM /flats/ua_38/warehouse_38_1.dat OF DEL MODIFIED BY
COLDEL| KEEPBLANKS COMPOUND=50 ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO WAREHOUSE38;
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 reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.CUSTOMER15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;

```



```

connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE26 AND INDEXES ALL;
COMMIT WORK;

```

```

connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE28 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.WAREHOUSE38 AND INDEXES ALL;
COMMIT WORK;
connect reset;

```

SORT ORDERS.sh

```

sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_01/orders_01_1.dat >
/flats/ua_00_01/orders_01_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_01/orders_01_1.dat.sorted /flats/ua_00_01/orders_01_1.dat
else

```

```

echo "**** WARNING ****"
echo "Sort of /flats/ua_00_01/orders_01_1.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_01/orders_01_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_01/orders_01_2.dat >
/flats/ua_00_01/orders_01_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_01/orders_01_2.dat.sorted /flats/ua_00_01/orders_01_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ua_00_01/orders_01_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_01/orders_01_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_02/orders_02_1.dat >
/flats/ua_00_02/orders_02_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_02/orders_02_1.dat.sorted /flats/ua_00_02/orders_02_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ua_00_02/orders_02_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_02/orders_02_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_02/orders_02_2.dat >
/flats/ua_00_02/orders_02_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_02/orders_02_2.dat.sorted /flats/ua_00_02/orders_02_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ua_00_02/orders_02_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_02/orders_02_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_03/orders_03_1.dat >
/flats/ua_00_03/orders_03_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_03/orders_03_1.dat.sorted /flats/ua_00_03/orders_03_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ua_00_03/orders_03_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_03/orders_03_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_03/orders_03_2.dat >
/flats/ua_00_03/orders_03_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_03/orders_03_2.dat.sorted /flats/ua_00_03/orders_03_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ua_00_03/orders_03_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_03/orders_03_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_04/orders_04_1.dat >
/flats/ua_00_04/orders_04_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_04/orders_04_1.dat.sorted /flats/ua_00_04/orders_04_1.dat

```



```

else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_04/orders_04_1.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_04/orders_04_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_04/orders_04_2.dat >
/flats/ua_00_04/orders_04_2.dat.sorted

if [ $? = 0 ]
then
mv /flats/ua_00_04/orders_04_2.dat.sorted /flats/ua_00_04/orders_04_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_04/orders_04_2.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_04/orders_04_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_05/orders_05_1.dat >
/flats/ua_00_05/orders_05_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_05/orders_05_1.dat.sorted /flats/ua_00_05/orders_05_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_05/orders_05_1.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_05/orders_05_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_05/orders_05_2.dat >
/flats/ua_00_05/orders_05_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_05/orders_05_2.dat.sorted /flats/ua_00_05/orders_05_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_05/orders_05_2.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_05/orders_05_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_06/orders_06_1.dat >
/flats/ua_00_06/orders_06_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_06/orders_06_1.dat.sorted /flats/ua_00_06/orders_06_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_06/orders_06_1.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_06/orders_06_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_06/orders_06_2.dat >
/flats/ua_00_06/orders_06_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_06/orders_06_2.dat.sorted /flats/ua_00_06/orders_06_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_06/orders_06_2.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_06/orders_06_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_07/orders_07_1.dat >
/flats/ua_00_07/orders_07_1.dat.sorted
if [ $? = 0 ]
then

```

```

mv /flats/ua_00_07/orders_07_1.dat.sorted /flats/ua_00_07/orders_07_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_07/orders_07_1.dat FAILED."

echo "Please ensure that the source file (/flats/ua_00_07/orders_07_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_07/orders_07_2.dat >
/flats/ua_00_07/orders_07_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_07/orders_07_2.dat.sorted /flats/ua_00_07/orders_07_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_07/orders_07_2.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_07/orders_07_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_08/orders_08_1.dat >
/flats/ua_00_08/orders_08_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_08/orders_08_1.dat.sorted /flats/ua_00_08/orders_08_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_08/orders_08_1.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_08/orders_08_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_08/orders_08_2.dat >
/flats/ua_00_08/orders_08_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_08/orders_08_2.dat.sorted /flats/ua_00_08/orders_08_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_08/orders_08_2.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_08/orders_08_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_09/orders_09_1.dat >
/flats/ua_00_09/orders_09_1.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_09/orders_09_1.dat.sorted /flats/ua_00_09/orders_09_1.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_09/orders_09_1.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_09/orders_09_1.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_09/orders_09_2.dat >
/flats/ua_00_09/orders_09_2.dat.sorted
if [ $? = 0 ]
then
mv /flats/ua_00_09/orders_09_2.dat.sorted /flats/ua_00_09/orders_09_2.dat
else
echo "***** WARNING *****"
echo "Sort of /flats/ua_00_09/orders_09_2.dat FAILED."
echo "Please ensure that the source file (/flats/ua_00_09/orders_09_2.dat)"
echo "exists and that sufficient free space is available"
echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_10/orders_10_1.dat >
/flats/ua_00_10/orders_10_1.dat.sorted
if [ $? = 0 ]

```

```

then
  mv /flats/ua_00_10/orders_10_1.dat.sorted /flats/ua_00_10/orders_10_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_10/orders_10_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_10/orders_10_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_10/orders_10_2.dat >
/flats/ua_00_10/orders_10_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_10/orders_10_2.dat.sorted /flats/ua_00_10/orders_10_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_10/orders_10_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_10/orders_10_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_11/orders_11_1.dat >
/flats/ua_00_11/orders_11_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_11/orders_11_1.dat.sorted /flats/ua_00_11/orders_11_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_11/orders_11_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_11/orders_11_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_11/orders_11_2.dat >
/flats/ua_00_11/orders_11_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_11/orders_11_2.dat.sorted /flats/ua_00_11/orders_11_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_11/orders_11_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_11/orders_11_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_12/orders_12_1.dat >
/flats/ua_00_12/orders_12_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_12/orders_12_1.dat.sorted /flats/ua_00_12/orders_12_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_12/orders_12_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_12/orders_12_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_12/orders_12_2.dat >
/flats/ua_00_12/orders_12_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_12/orders_12_2.dat.sorted /flats/ua_00_12/orders_12_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_12/orders_12_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_12/orders_12_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_13/orders_13_1.dat >
/flats/ua_00_13/orders_13_1.dat.sorted
if [ $? = 0 ]

```

```

then
  mv /flats/ua_00_13/orders_13_1.dat.sorted /flats/ua_00_13/orders_13_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_13/orders_13_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_13/orders_13_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_13/orders_13_2.dat >
/flats/ua_00_13/orders_13_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_13/orders_13_2.dat.sorted /flats/ua_00_13/orders_13_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_13/orders_13_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_13/orders_13_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_14/orders_14_1.dat >
/flats/ua_00_14/orders_14_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_14/orders_14_1.dat.sorted /flats/ua_00_14/orders_14_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_14/orders_14_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_14/orders_14_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_14/orders_14_2.dat >
/flats/ua_00_14/orders_14_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_14/orders_14_2.dat.sorted /flats/ua_00_14/orders_14_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_14/orders_14_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_14/orders_14_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_15/orders_15_1.dat >
/flats/ua_00_15/orders_15_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_15/orders_15_1.dat.sorted /flats/ua_00_15/orders_15_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_15/orders_15_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_15/orders_15_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_15/orders_15_2.dat >
/flats/ua_00_15/orders_15_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_15/orders_15_2.dat.sorted /flats/ua_00_15/orders_15_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_15/orders_15_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_15/orders_15_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_16/orders_16_1.dat >
/flats/ua_00_16/orders_16_1.dat.sorted
if [ $? = 0 ]

```

```

then
  mv /flats/ua_00_16/orders_16_1.dat.sorted /flats/ua_00_16/orders_16_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_16/orders_16_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_16/orders_16_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_16/orders_16_2.dat >
/flats/ua_00_16/orders_16_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_16/orders_16_2.dat.sorted /flats/ua_00_16/orders_16_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_16/orders_16_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_16/orders_16_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_17/orders_17_1.dat >
/flats/ua_00_17/orders_17_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_17/orders_17_1.dat.sorted /flats/ua_00_17/orders_17_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_17/orders_17_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_17/orders_17_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_17/orders_17_2.dat >
/flats/ua_00_17/orders_17_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_17/orders_17_2.dat.sorted /flats/ua_00_17/orders_17_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_17/orders_17_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_17/orders_17_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_18/orders_18_1.dat >
/flats/ua_00_18/orders_18_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_18/orders_18_1.dat.sorted /flats/ua_00_18/orders_18_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_18/orders_18_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_18/orders_18_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_18/orders_18_2.dat >
/flats/ua_00_18/orders_18_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_18/orders_18_2.dat.sorted /flats/ua_00_18/orders_18_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_18/orders_18_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_18/orders_18_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_19/orders_19_1.dat >
/flats/ua_00_19/orders_19_1.dat.sorted

```

```

if [ $? = 0 ]
then
  mv /flats/ua_00_19/orders_19_1.dat.sorted /flats/ua_00_19/orders_19_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_19/orders_19_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_19/orders_19_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_19/orders_19_2.dat >
/flats/ua_00_19/orders_19_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_19/orders_19_2.dat.sorted /flats/ua_00_19/orders_19_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_19/orders_19_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_19/orders_19_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_20/orders_20_1.dat >
/flats/ua_00_20/orders_20_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_20/orders_20_1.dat.sorted /flats/ua_00_20/orders_20_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_20/orders_20_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_20/orders_20_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_20/orders_20_2.dat >
/flats/ua_00_20/orders_20_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_20/orders_20_2.dat.sorted /flats/ua_00_20/orders_20_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_20/orders_20_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_20/orders_20_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_21/orders_21_1.dat >
/flats/ua_00_21/orders_21_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_21/orders_21_1.dat.sorted /flats/ua_00_21/orders_21_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_21/orders_21_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_21/orders_21_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_21/orders_21_2.dat >
/flats/ua_00_21/orders_21_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_21/orders_21_2.dat.sorted /flats/ua_00_21/orders_21_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_21/orders_21_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_21/orders_21_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_22/orders_22_1.dat >
/flats/ua_00_22/orders_22_1.dat.sorted

```

```

if [ $? = 0 ]
then
  mv /flats/ua_00_22/orders_22_1.dat.sorted /flats/ua_00_22/orders_22_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_22/orders_22_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_22/orders_22_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_22/orders_22_2.dat >
/flats/ua_00_22/orders_22_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_22/orders_22_2.dat.sorted /flats/ua_00_22/orders_22_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_22/orders_22_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_22/orders_22_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_23/orders_23_1.dat >
/flats/ua_00_23/orders_23_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_23/orders_23_1.dat.sorted /flats/ua_00_23/orders_23_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_23/orders_23_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_23/orders_23_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_23/orders_23_2.dat >
/flats/ua_00_23/orders_23_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_23/orders_23_2.dat.sorted /flats/ua_00_23/orders_23_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_23/orders_23_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_23/orders_23_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_24/orders_24_1.dat >
/flats/ua_00_24/orders_24_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_24/orders_24_1.dat.sorted /flats/ua_00_24/orders_24_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_24/orders_24_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_24/orders_24_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_24/orders_24_2.dat >
/flats/ua_00_24/orders_24_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_24/orders_24_2.dat.sorted /flats/ua_00_24/orders_24_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_24/orders_24_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_24/orders_24_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_25/orders_25_1.dat >
/flats/ua_00_25/orders_25_1.dat.sorted

```

```

if [ $? = 0 ]
then
  mv /flats/ua_00_25/orders_25_1.dat.sorted /flats/ua_00_25/orders_25_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_25/orders_25_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_25/orders_25_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_25/orders_25_2.dat >
/flats/ua_00_25/orders_25_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_25/orders_25_2.dat.sorted /flats/ua_00_25/orders_25_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_25/orders_25_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_25/orders_25_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_26/orders_26_1.dat >
/flats/ua_00_26/orders_26_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_26/orders_26_1.dat.sorted /flats/ua_00_26/orders_26_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_26/orders_26_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_26/orders_26_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_26/orders_26_2.dat >
/flats/ua_00_26/orders_26_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_26/orders_26_2.dat.sorted /flats/ua_00_26/orders_26_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_26/orders_26_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_26/orders_26_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_27/orders_27_1.dat >
/flats/ua_00_27/orders_27_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_27/orders_27_1.dat.sorted /flats/ua_00_27/orders_27_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_27/orders_27_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_27/orders_27_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_27/orders_27_2.dat >
/flats/ua_00_27/orders_27_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_27/orders_27_2.dat.sorted /flats/ua_00_27/orders_27_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_27/orders_27_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_27/orders_27_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi

```

```

sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_28/orders_28_1.dat >
/flats/ua_00_28/orders_28_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_28/orders_28_1.dat.sorted /flats/ua_00_28/orders_28_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_28/orders_28_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_28/orders_28_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_28/orders_28_2.dat >
/flats/ua_00_28/orders_28_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_28/orders_28_2.dat.sorted /flats/ua_00_28/orders_28_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_28/orders_28_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_28/orders_28_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_29/orders_29_1.dat >
/flats/ua_00_29/orders_29_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_29/orders_29_1.dat.sorted /flats/ua_00_29/orders_29_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_29/orders_29_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_29/orders_29_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_29/orders_29_2.dat >
/flats/ua_00_29/orders_29_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_29/orders_29_2.dat.sorted /flats/ua_00_29/orders_29_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_29/orders_29_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_29/orders_29_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_30/orders_30_1.dat >
/flats/ua_00_30/orders_30_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_30/orders_30_1.dat.sorted /flats/ua_00_30/orders_30_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_30/orders_30_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_30/orders_30_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_30/orders_30_2.dat >
/flats/ua_00_30/orders_30_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_30/orders_30_2.dat.sorted /flats/ua_00_30/orders_30_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_30/orders_30_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_30/orders_30_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi

```

```

sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_31/orders_31_1.dat >
/flats/ua_00_31/orders_31_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_31/orders_31_1.dat.sorted /flats/ua_00_31/orders_31_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_31/orders_31_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_31/orders_31_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_31/orders_31_2.dat >
/flats/ua_00_31/orders_31_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_31/orders_31_2.dat.sorted /flats/ua_00_31/orders_31_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_31/orders_31_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_31/orders_31_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_32/orders_32_1.dat >
/flats/ua_00_32/orders_32_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_32/orders_32_1.dat.sorted /flats/ua_00_32/orders_32_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_32/orders_32_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_32/orders_32_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_32/orders_32_2.dat >
/flats/ua_00_32/orders_32_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_32/orders_32_2.dat.sorted /flats/ua_00_32/orders_32_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_32/orders_32_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_32/orders_32_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_33/orders_33_1.dat >
/flats/ua_00_33/orders_33_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_33/orders_33_1.dat.sorted /flats/ua_00_33/orders_33_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_33/orders_33_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_33/orders_33_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_33/orders_33_2.dat >
/flats/ua_00_33/orders_33_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_33/orders_33_2.dat.sorted /flats/ua_00_33/orders_33_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_33/orders_33_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_33/orders_33_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi

```

```

sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_34/orders_34_1.dat >
/flats/ua_00_34/orders_34_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_34/orders_34_1.dat.sorted /flats/ua_00_34/orders_34_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_34/orders_34_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_34/orders_34_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_34/orders_34_2.dat >
/flats/ua_00_34/orders_34_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_34/orders_34_2.dat.sorted /flats/ua_00_34/orders_34_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_34/orders_34_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_34/orders_34_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_35/orders_35_1.dat >
/flats/ua_00_35/orders_35_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_35/orders_35_1.dat.sorted /flats/ua_00_35/orders_35_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_35/orders_35_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_35/orders_35_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_35/orders_35_2.dat >
/flats/ua_00_35/orders_35_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_35/orders_35_2.dat.sorted /flats/ua_00_35/orders_35_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_35/orders_35_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_35/orders_35_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_36/orders_36_1.dat >
/flats/ua_00_36/orders_36_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_36/orders_36_1.dat.sorted /flats/ua_00_36/orders_36_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_36/orders_36_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_36/orders_36_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_36/orders_36_2.dat >
/flats/ua_00_36/orders_36_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_36/orders_36_2.dat.sorted /flats/ua_00_36/orders_36_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_36/orders_36_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_36/orders_36_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi

```

```

sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_37/orders_37_1.dat >
/flats/ua_00_37/orders_37_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_37/orders_37_1.dat.sorted /flats/ua_00_37/orders_37_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_37/orders_37_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_37/orders_37_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_37/orders_37_2.dat >
/flats/ua_00_37/orders_37_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_37/orders_37_2.dat.sorted /flats/ua_00_37/orders_37_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_37/orders_37_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_37/orders_37_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_38/orders_38_1.dat >
/flats/ua_00_38/orders_38_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_38/orders_38_1.dat.sorted /flats/ua_00_38/orders_38_1.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_38/orders_38_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_38/orders_38_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
fi
sort -t'|' -T /ixtemp +5n +6n +7n /flats/ua_00_38/orders_38_2.dat >
/flats/ua_00_38/orders_38_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ua_00_38/orders_38_2.dat.sorted /flats/ua_00_38/orders_38_2.dat
else
  echo "***** WARNING *****"
  echo "Sort of /flats/ua_00_38/orders_38_2.dat FAILED."
  echo "Please ensure that the source file (/flats/ua_00_38/orders_38_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/ixtemp)."
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$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

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

LDFLAGS = $(LD_FLAGS_EXEC) $(LD_FLAGS_LIB)

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

OBJS = tpcrnd$(OBJEXT) \
$(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT)
OBJ_EEE = $(TPCC_ROOT)/Src.Common/tpcc1wh$(OBJEXT)

EXEC = gendata$(BINEXT)

#
#####
#####
# End-User Targets
#
#####
#####

all: $(EXEC)

clean:
- $(ERASE) *$(OBJEXT) $(EXEC)

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

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

# We use @$$(OBJEXT) here so that the UNIX makefiles work with both
# 'traditional' make and GNU make
$(EXEC):
$(LD_EXEC) $(LD_FLAGS) $(OBJS) @$$(OBJEXT)
$(LD_FLAGS_OUT)$@

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

# Link Dependencies
gendata$(BINEXT): $(OBJS) gendata$(OBJEXT)

```

```

# Build Dependencies
# Source
gendata$(OBJEXT): gendata.c

# Headers
gendata.c: $(TPCC_ROOT)/include/tpccrnd.h $(TPCC_ROOT)/include/lval.h

dbgen/gendata.c

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

/*
* gendata.c - Generate data for TPC-C database
*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sqlutil.h>
/* UNIX named pipe support */
#include <sys/stat.h>
#include <errno.h>
#include <fcntl.h>

#include "platform.h"
#include "tpccrnd.h"
#include "lval.h"

/* PROTOTYPES. */
void gen_dist_tbl( void );
void gen_cust_tbl( void );
void gen_hist_tbl( void );
void gen_nu_ord_tbl( void );
void gen_ordr_tbl( void );
void gen_item_tbl( void );
void gen_stock_tbl( void );
void gen_ware_tbl( void );

int i, j;
double timestamp1, timestamp2, elapse;
int rc, rc1, rc2;

int using_range = 0;
int using_npipe = 0;
int using_rctload = 0;
int quiet_mode = 0;
sqlint32 ware_start=-1, ware_end=-1;

char fmtWare[] = "%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s\n";
char fmtDist[] = "%d|%d|%d|%d|%s|%s|%s|%s|%s|%s|%s|%s\n";
char fmtItem[] = "%s|%d|%s|%d|%s\n";
char fmtStock[] =
"%d|%d|%d|%d|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s\n";
char fmtCust[] =
"%d|%s|%s|%s|%s|%d|%d|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%d|%d|%d|%d|%d|%d\n";
char fmtHist[] = "%d|%d|%d|%d|%d|%d|%d|%d|%s\n";

```

```

char fmtOrdrr[] = "%d%lld%d%d%d%d%d%d\n";
char fmtOLine[] = "%lld%d%d%d%d%d%d%d%d%d\n";
char fmtNewOrd[] = "%d%d%d\n";
void InitFormatStrings(char delim);
void ScalingReport(void);

int outtype1 = 0;
int outtype2 = 0;
char *outname1 = NULL;
char *outname2 = NULL;

/*-----*/
/* main */
/*-----*/
int main (int argc, char *argv[])
{
    int option = -1;
    char *delim = NULL;

    /*
    *****
    */
    /* Compute Warehouse Ranges */
    /*
    *****
    */
    ware_start = 1;
    ware_end = WAREHOUSES;

    /*
    *****
    */
    /* Process Command Line Arguments */
    /*
    *****
    */

    /* Valid Command Line Options
    * -----
    * Table Option:      -t <table>      (-t 3 for warehouse)
    * Output Column Delimiter: -d <char>      (-d '|', -d '\', etc)
    * Output to File:      -f[n] <file>      (-f customer.dat)
    * Output to Pipe:      -p[n] <pipename>  (-p tpccpipe.000)
    * Warehouse Range:    -r <start> <end>  (-r 1 100)
    * Scaling Report:     -s
    * Quiet Mode:         -q
    *
    * The -f[n] and/or -p[n] options are required.
    * The -t, -d, -r, -s and -q options are optional.
    *
    * If -d is omitted, the vertical bar (pipe) symbol (|) will be used.
    * If -r is omitted, the range [1..WAREHOUSES] will be used.
    *
    * Due to the TPC-C spec requiring that orders and orderline be
    * generated at the same time, there is an extension to the -f and -p
    * options to specify one of the two output streams for each argument.
    *
    * -f1 orders.dat -f2 orderline.dat will output to two files
    * -f1 orders.dat -p2 tpccpipe.000 will output to a file and a pipe
    *
    * -f1/-p1 specifies the destination for the orders table
    * -f2/-p2 specifies the destination for the orderline table
    *
    */

    /* Read Arguments */
    for (i=1; i<argc; i++)
    {
        if (strcmp(argv[i], "-t") == 0) {
            option = atoi(argv[i+1]);
            i++;
        } else if (strcmp(argv[i], "-r") == 0) {

```

```

        ware_start = atoi(argv[i+1]);
        ware_end = atoi(argv[i+2]);
        i += 2;
    } else if (strcmp(argv[i], "-d") == 0) {
        delim = argv[i+1];
        i++;
    } else if ((strcmp(argv[i], "-f") == 0) ||
                (strcmp(argv[i], "-f1") == 0)) {
        outtype1 = IOH_FILE;
        outname1 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-f2") == 0) {
        outtype2 = IOH_FILE;
        outname2 = argv[i+1];
        i++;
    } else if ((strcmp(argv[i], "-p") == 0) ||
                (strcmp(argv[i], "-p1") == 0)) {
        outtype1 = IOH_PIPE;
        outname1 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-p2") == 0) {
        outtype2 = IOH_PIPE;
        outname2 = argv[i+1];
        i++;
    } else if (strcmp(argv[i], "-s") == 0) {
        ScalingReport();
        exit(0);
    } else if (strcmp(argv[i], "-q") == 0) {
        quiet_mode = 1;
    } else {
        fprintf(stderr, "gendata: Don't understand argument: %s\n", argv[i]);
        exit(-1);
    }
}

/*
*****
*/
/* Validate Command Line Arguments */
/*
*****
*/

/* Validate Table Argument */
if (option < 3 || option > 11 || option == 10)
{
    fprintf(stderr, "gendata: Invalid table selected: %d\n", option);
    exit(-1);
}

/* Validate Delimiter Argument */
if (delim == NULL) {
    // default delimiter is used for IMPORT & LOAD, no changes 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();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nITEM table generated in %8.2f seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nITEM table FAILED at (I %d) after %8.2f
seconds.\n\n", item_num, elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate stock table */
/*-----*/

void gen_stock_tbl( void )
{
    sqlint32 ware_num = 0 ;

```

```

sqlint32 stock_num = 0 ;
sqlint32 stock_quant;
sqlint32 s_ytd;
sqlint32 s_order_cnt, s_remote_cnt;
char stock_dist_01[25] ;
char stock_dist_02[25] ;
char stock_dist_03[25] ;
char stock_dist_04[25] ;
char stock_dist_05[25] ;
char stock_dist_06[25] ;
char stock_dist_07[25] ;
char stock_dist_08[25] ;
char stock_dist_09[25] ;
char stock_dist_10[25] ;
char stock_data[51] ;

IOH_NUM numBytes;
ioHandle hnd;
char Buffer[1024];

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();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nSTOCK table generated in %8.2f seconds.\n\n",elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nSTOCK table FAILED at (S %d W %d) after %8.2f
seconds.\n\n",stock_num, ware_num, elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate warehouse table */
/*-----*/

void gen_ware_tbl( void )
{
    sqlint32 ware_num = 0 ;
    char ware_name[11] ;
    char ware_street_1[21] ;
    char ware_street_2[21] ;
    char ware_city[21] ;
    char ware_state[3] ;
    char ware_zip[10] ;
    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)) { // @dxxxxxmtc
            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();
    elapse = timestamp2 - timestamp1;
    if (rc == 0) {
        fprintf(stdout, "\nWAREHOUSE table generated in %8.2f
seconds.\n\n", elapse);
        fflush(stdout);
    } else {
        fprintf(stderr, "\nWAREHOUSE table FAILED at (W %d) after %8.2f
seconds.\n\n", ware_num, elapse);
        fflush(stderr);
    }
}

/*-----*/
/* generate dist table */
/*-----*/

void gen_dist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    char dist_name[11];
    char dist_street_1[21];
    char dist_street_2[21];
    char dist_city[21];
    char dist_state[3];
    char dist_zip[10];
    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 currmtstmp;
    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-rectload) */
            if (using_rectload == 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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nCUSTOMER table generated in %8.2f
seconds.\n\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nCUSTOMER table FAILED at (W %d D %d C %d) after
%8.2f seconds.\n\n", ware_num, dist_num, cust_num, elapsed);
    fflush(stderr);
}
}

/*-----*/
/* generate hist table */
/*-----*/

void gen_hist_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    char hist_data[25] ;
    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();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nHISTORY table generated in %8.2f seconds.\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nHISTORY table FAILED at (W %d D %d C %d) after
%8.2f seconds.\n", ware_num, dist_num, cust_num, elapsed);
    fflush(stderr);
}
}

/*-----*/
/* generate nu_ord table */
/*-----*/
void gen_nu_ord_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 nu_ord_id = 0 ;
    int nu_ord_hi ;

    IOH_NUM numBytes;
    ioHandle hnd;
    char Buffer[1024];

    /* compute maximum and minimum
    order numbers for this
    district */
    nu_ord_hi = CUSTOMERS_PER_DISTRICT -
    NU_ORDERS_PER_DISTRICT + 1;
    if (nu_ord_hi < 0) {
        nu_ord_hi = CUSTOMERS_PER_DISTRICT -
    (CUSTOMERS_PER_DISTRICT / 3) + 1;
        fprintf(stderr, "\n**** WARNING **** NU_ORDERS_PER_DISTRICT is
> CUSTOMERS_PER_DISTRICT\n");
        fprintf(stderr, "        Check the values in file lval.h\n");
        fprintf(stderr, "        Loading New-Order with 1/3 of
CUSTOMERS_PER_DISTRICT\n");
    }
    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... */
    } /* end for... */

    rc = GenericClose(&hnd);

neword_done:

timestamp2 = current_time();
elapsed = timestamp2 - timestamp1;
if (rc == 0) {
    fprintf(stdout, "\nNEW_ORDER table generated in %8.2f
seconds.\n", elapsed);
    fflush(stdout);
} else {
    fprintf(stderr, "\nNEW_ORDER table FAILED at (W %d D %d O %d) after
%8.2f seconds.\n", ware_num, dist_num, nu_ord_id, elapsed);
    fflush(stderr);
}
}

/*-----*/
/* generate order and order_line tables */
/*-----*/
void gen_ordr_tbl( void )
{
    sqlint32 ware_num = 0 ;
    sqlint32 dist_num = 0 ;
    sqlint32 cust_num = 0 ;
    sqlint32 ord_num = 0 ;
    sqlint32 ordr_carrier_id;
    sqlint32 ordr_ol_cnt;
    sqlint32 oline_ol_num;
    sqlint32 oline_item_num;

    sqlint32 oline_amount;
    char oline_dist_info[25];
    sqlint64 nulltmstmp = 0;
    sqlint64 currtmstmp;

    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; }

    currtmstmp = 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)
      ord_carrier_id = rand_integer( 1, 10 );
    else
      ord_carrier_id = 0;

    cust_num = random_1_3000();
    ord_ol_cnt =
rand_integer(MIN_OL_PER_ORDER,MAX_OL_PER_ORDER);

    numBytes = sprintf(Buffer, fmtOrdr,
      cust_num,
      currtmstp,
      ord_carrier_id,
      ord_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 <= ord_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();
elapsed = timestamp2 - timestamp1;
if (rc1 == 0 && rc2 == 0) {
  fprintf(stdout, "\nORDERS & ORDER_LINE table(s) generated in %8.2f
seconds.\n\n", elapsed);
  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, elapsed);
  fflush(stderr);
}
}

// This routine will initialize the printf format strings and replace the
// delimiter with the one provided. The pipe symbol is the default.
void InitFormatStrings(char delim)
{
  char *p;

  // Check if Using Default Delimiter
  if (delim == '|') return;

  // Replace Delimiters
  while (p = strchr(fmtWare, '|')) { *p = delim; }
  while (p = strchr(fmtDist, '|')) { *p = delim; }
  while (p = strchr(fmtItem, '|')) { *p = delim; }
  while (p = strchr(fmtStock, '|')) { *p = delim; }
  while (p = strchr(fmtCust, '|')) { *p = delim; }
  while (p = strchr(fmtHist, '|')) { *p = delim; }
  while (p = strchr(fmtOrdr, '|')) { *p = delim; }
  while (p = strchr(fmtOLine, '|')) { *p = delim; }
  while (p = strchr(fmtNewOrd, '|')) { *p = delim; }
}

void ScalingReport(void)
{
  /* Print Scaling Values */
  fprintf(stdout, "Scaling Values in Use\n");
  fprintf(stdout, "-----\n");
  fprintf(stdout, "Warehouses:      %d\n", WAREHOUSES);
  fprintf(stdout, "Districts/Warehouse:  %d\n",
DISTRICTS_PER_WAREHOUSE);
  fprintf(stdout, "Customers/District:  %d\n",
CUSTOMERS_PER_DISTRICT);
  fprintf(stdout, "Items:                %d\n", ITEMS);
  fprintf(stdout, "Stock/Warehouse:     %d\n", STOCK_PER_WAREHOUSE);
  fprintf(stdout, "Min Order Lines/Order: %d\n", MIN_OL_PER_ORDER);
  fprintf(stdout, "Max Order Lines/Order: %d\n", MAX_OL_PER_ORDER);
  fprintf(stdout, "New Orders/District: %d\n",
NU_ORDERS_PER_DISTRICT);
  fprintf(stdout, "-----\n");
}

```

dbgen/tpccrnd.c

```

/*****
*****
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 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 alnum[] =
"0123456789abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ
WXYZ";

static char *last_name_parts[] =
{
"BAR",
"OUGHT",
"ABLE",
"PRI",
"PRES",
"ESE",
"ANTI",
"CALLY",
"ATION",
"EING"
};

/*
*****
*****
* seed_1_3000
*
*
*****
*****
*/

void seed_1_3000( void )
{
int i;

for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++) {
tbl_cust[i] = 0;
}
}

/*
*****
*****
* random_1_3000
*
*
*****
*****
*/

int random_1_3000( void )
{
static int i;
static int x;

x = rand_integer(0, CUSTOMERS_PER_DISTRICT - 1);

for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++)
{
if (tbl_cust[x] == 0)
{
tbl_cust[x] = 1;
return(x+1);
} else {
x++;
}
}
if (x == CUSTOMERS_PER_DISTRICT)
x=0;
}

printf("\nfatal error in random_1_3000 \n");
abort();
}

```

```

/*
*****
*****
* initialize_random
*
*
*****
*****
*/

void initialize_random(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 81700
#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
*/

#ifndef __PLATFORM_H
#define __PLATFORM_H

/*
*****
*****/

/* Generic Macros */
/*
*****
*****/

```

```

#define GEN_ERRCODE errno

/*
*****
*****/

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

/*
*****
*****/

/* UNIX I/O Macros */
/*
*****
*****/

#include <fcntl.h>

#define IOH_INIT(hnd, type, name)
    hnd->fd = -1;
    hnd->type = type;
    hnd->name = name;

#define IOH_CREATE(hnd)
    if (hnd->type == IOH_PIPE) {
        rc = mkfifo(hnd->name, 0666);
    } else {
        rc = 0;
    }

#define IOH_OPEN(hnd)
    if (hnd->type == IOH_FILE_APPEND) {
        hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_APPEND,
0666);
    } else {
        hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_TRUNC, 0666);
    }
    if (hnd->fd == -1) {
        rc = -1;
    } else {
        rc = 0;
    }

#define IOH_WRITE(hnd, buff, num, num2)
    rc = write(hnd->fd, buff, num);
    if (rc >= 0) {
        num2 = rc;
        rc = 0;
    }

#define IOH_FLUSH(hnd) rc = 0;
#define IOH_CLOSE(hnd) rc = close(hnd->fd);
#define IOH_DELETE(hnd) if (hnd->type == IOH_PIPE) { rc = unlink(hnd->name); }

typedef unsigned int IOH_NUM;
typedef int IOH_HND;

/*
*****
*****/

/* UNIX Semaphore Macros */
/*
*****
*****/

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

union semun {
    int val;
    struct semid_ds *buf;

```

```

unsigned short int *array;
} semUnion;

struct sembuf semBuf;

#define SEM_HANDLE int

#define SEM_INIT(hnd, x, name)
if ( (hnd = semget(IPC_PRIVATE, 1, IPC_CREAT | IPC_EXCL | S_IRUSR
| S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH)) == -1)
\
API_ERROR(__LINE__, "semget", (rc=GEN_ERRCODE));
semUnion.val = x;
if ( semctl(hnd, 0, SETVAL, semUnion) < 0 )
API_ERROR(__LINE__, "semctl SETVAL", (rc=GEN_ERRCODE));

#define SEM_WAIT(hnd)
semBuf.sem_num = 0;
semBuf.sem_op = -1;
semBuf.sem_flg = SEM_UNDO;
if ( semop(hnd, &semBuf, 1) < 0 )
API_ERROR(__LINE__, "semop wait", (rc=GEN_ERRCODE));

#define SEM_FREE(hnd)
semBuf.sem_num = 0;
semBuf.sem_op = 1;
semBuf.sem_flg = SEM_UNDO;
if ( semop(hnd, &semBuf, 1) < 0 )
API_ERROR(__LINE__, "semop free", (rc=GEN_ERRCODE));

#define SEM_DESTROY(hnd)
if ( semctl(hnd, 0, IPC_RMID, 0) )
API_ERROR(__LINE__, "semctl IPC_RMID", (rc=GEN_ERRCODE));

/*
*****
*****/
/* Common I/O Macros and Definitions */
/*
*****
*****/
#define IOH_FILE 1
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3

#define IOH_ERRMSG(hnd, msg)
if (rc != 0) {
fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", GEN_ERRCODE, msg,
hnd->fd, hnd->type, hnd->name);
return rc;
}

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

typedef struct _ioh ioHandle;

/*
*****
*****/
/* Generic I/O Routine Prototypes */
/*
*****
*****/
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes);
int GenericClose(ioHandle *hnd);

```

```

/*
*****
*****/
/* Generic I/O Routines */
/*
*****
*****/
int GenericOpen(ioHandle *hnd, int type, char *name)
{
int rc = 0;

IOH_INIT(hnd, type, name)

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

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

return rc;
}

int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes)
{
int rc = 0;
int numBytesWritten = -1;

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

int GenericClose(ioHandle *hnd)
{
int rc = 0;

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

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

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

return rc;
}

#endif // __PLATFORM_H

include/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 NUrnd_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

```
#define MASTER_NUM1 1
#define MASTER_NUM2 0
#define MASTER_NUM3 0
#define MASTER_NUM4 0
#define MASTER_NUM5 0
#define MASTER_NUM6 0
#define MASTER_NUM7 0
#define MASTER_NUM8 0
#define MASTER_NUM9 0
#define MASTER_NUM10 0
```

```
#if MASTER_NUM1
MASTER "master1"
#elif MASTER_NUM2
MASTER "master2"
#elif MASTER_NUM3
MASTER "master3"
#elif MASTER_NUM4
MASTER "master4"
#elif MASTER_NUM5
MASTER "master5"
#elif MASTER_NUM6
MASTER "master6"
#elif MASTER_NUM7
MASTER "master7"
#elif MASTER_NUM8
MASTER "master8"
#elif MASTER_NUM9
MASTER "master9"
#elif MASTER_NUM10
MASTER "master10"
#endif
```

```
/*---- SUT -----*/
SUT="toraus"
/*-----*/
LASTC=88
```

```
MEASUREMENT="1"
WAREHOUSES=81700
/*----- SLAVES -----*/
```

```
#if MASTER_NUM1
SLAVES driver1a, driver1b, driver1c, driver1d, driver2a, driver2b, driver2c,
driver2d, driver3a, driver3b, driver3c, driver3d, driver4a, driver4b, driver4c,
driver4d
#elif MASTER_NUM2
SLAVES driver5a, driver5b, driver5c, driver5d, driver6a, driver6b, driver6c,
driver6d, driver7a, driver7b, driver7c, driver7d, driver8a, driver8b, driver8c,
driver8d
#elif MASTER_NUM3
SLAVES driver9a, driver9b, driver9c, driver9d, driver10a, driver10b, driver10c,
driver10d, driver11a, driver11b, driver11c, driver11d, driver12a, driver12b,
driver12c, driver12d
#elif MASTER_NUM4
SLAVES driver13a, driver13b, driver13c, driver13d, driver14a, driver14b,
driver14c, driver14d, driver15a, driver15b, driver15c, driver15d, driver16a,
driver16b, driver16c, driver16d
#elif MASTER_NUM5
SLAVES driver17a, driver17b, driver17c, driver17d, driver18a, driver18b,
driver18c, driver18d, driver19a, driver19b, driver19c, driver19d, driver20a,
driver20b, driver20c, driver20d
#elif MASTER_NUM6
```

```
SLAVES driver21a, driver21b, driver21c, driver21d, driver22a, driver22b,
driver22c, driver22d, driver23a, driver23b, driver23c, driver23d, driver24a,
driver24b, driver24c, driver24d
#elif MASTER_NUM7
SLAVES driver25a, driver25b, driver25c, driver25d, driver26a, driver26b,
driver26c, driver26d, driver27a, driver27b, driver27c, driver27d, driver28a,
driver28b, driver28c, driver28d
#elif MASTER_NUM8
SLAVES driver29a, driver29b, driver29c, driver29d, driver30a, driver30b,
driver30c, driver30d, driver31a, driver31b, driver31c, driver31d, driver32a,
driver32b, driver32c, driver32d
#elif MASTER_NUM9
SLAVES driver33a, driver33b, driver33c, driver33d, driver34a, driver34b,
driver34c, driver34d, driver35a, driver35b, driver35c, driver35d, driver36a,
driver36b, driver36c, driver36d
#elif MASTER_NUM10
SLAVES driver37a, driver37b, driver37c, driver37d, driver38a, driver38b,
driver38c, driver38d, driver39a, driver39b, driver39c, driver39d, driver40a,
driver40b, driver40c, driver40d
#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"
#endif
```

```
/*---- more client ctuff -----*/
#if MASTER_NUM1
CLIENT client1a tpcc tpcc
CLIENT client1b tpcc tpcc

CLIENT client2a tpcc tpcc
CLIENT client2b tpcc tpcc

CLIENT client3a tpcc tpcc
CLIENT client3b tpcc tpcc

CLIENT client4a tpcc tpcc
CLIENT client4b tpcc tpcc

#elif MASTER_NUM2
CLIENT client5a tpcc tpcc
CLIENT client5b tpcc tpcc
```

```
CLIENT client6a tpcc tpcc
CLIENT client6b tpcc tpcc

CLIENT client7a tpcc tpcc
CLIENT client7b tpcc tpcc

CLIENT client8a tpcc tpcc
CLIENT client8b tpcc tpcc

#elif MASTER_NUM3
CLIENT client9a tpcc tpcc
CLIENT client9b tpcc tpcc

CLIENT client10a tpcc tpcc
CLIENT client10b tpcc tpcc

CLIENT client11a tpcc tpcc
CLIENT client11b tpcc tpcc

CLIENT client12a tpcc tpcc
CLIENT client12b tpcc tpcc

#elif MASTER_NUM4
CLIENT client13a tpcc tpcc
CLIENT client13b tpcc tpcc

CLIENT client14a tpcc tpcc
CLIENT client14b tpcc tpcc

CLIENT client15a tpcc tpcc
CLIENT client15b tpcc tpcc

CLIENT client16a tpcc tpcc
CLIENT client16b tpcc tpcc

#elif MASTER_NUM5
CLIENT client17a tpcc tpcc
CLIENT client17b tpcc tpcc

CLIENT client18a tpcc tpcc
CLIENT client18b tpcc tpcc

CLIENT client19a tpcc tpcc
CLIENT client19b tpcc tpcc

CLIENT client20a tpcc tpcc
CLIENT client20b tpcc tpcc

#elif MASTER_NUM6
CLIENT client21a tpcc tpcc
CLIENT client21b tpcc tpcc

CLIENT client22a tpcc tpcc
CLIENT client22b tpcc tpcc

CLIENT client23a tpcc tpcc
CLIENT client23b tpcc tpcc

CLIENT client24a tpcc tpcc
CLIENT client24b tpcc tpcc

#elif MASTER_NUM7
CLIENT client25a tpcc tpcc
CLIENT client25b tpcc tpcc

CLIENT client26a tpcc tpcc
CLIENT client26b tpcc tpcc

CLIENT client27a tpcc tpcc
CLIENT client27b tpcc tpcc
```

```
CLIENT client28a tpcc tpcc
CLIENT client28b tpcc tpcc

#elif MASTER_NUM8
CLIENT client29a tpcc tpcc
CLIENT client29b tpcc tpcc

CLIENT client30a tpcc tpcc
CLIENT client30b tpcc tpcc

CLIENT client31a tpcc tpcc
CLIENT client31b tpcc tpcc

CLIENT client32a tpcc tpcc
CLIENT client32b tpcc tpcc

#elif MASTER_NUM9
CLIENT client33a tpcc tpcc
CLIENT client33b tpcc tpcc

CLIENT client34a tpcc tpcc
CLIENT client34b tpcc tpcc

CLIENT client35a tpcc tpcc
CLIENT client35b tpcc tpcc

CLIENT client36a tpcc tpcc
CLIENT client36b tpcc tpcc

#elif MASTER_NUM10
CLIENT client37a tpcc tpcc
CLIENT client37b tpcc tpcc

CLIENT client38a tpcc tpcc
CLIENT client38b tpcc tpcc

CLIENT client39a tpcc tpcc
CLIENT client39b tpcc tpcc

CLIENT client40a tpcc tpcc
CLIENT client40b 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 socket1163 80 driver37c
SOCKET_NETWORK socket1164 80 driver37d
SOCKET_NETWORK socket1165 80 driver37a
SOCKET_NETWORK socket1166 80 driver37b
SOCKET_NETWORK socket1167 80 driver37c
SOCKET_NETWORK socket1168 80 driver37d
SOCKET_NETWORK socket1169 80 driver37a
SOCKET_NETWORK socket1170 80 driver37b
SOCKET_NETWORK socket1171 80 driver37c
SOCKET_NETWORK socket1172 80 driver37d
SOCKET_NETWORK socket1173 80 driver37a
SOCKET_NETWORK socket1174 80 driver37b
SOCKET_NETWORK socket1175 80 driver37c
SOCKET_NETWORK socket1176 80 driver37d
SOCKET_NETWORK socket1177 80 driver37a
SOCKET_NETWORK socket1178 80 driver37b
SOCKET_NETWORK socket1179 80 driver37c
SOCKET_NETWORK socket1180 80 driver37d
SOCKET_NETWORK socket1181 80 driver37a
SOCKET_NETWORK socket1182 80 driver37b
SOCKET_NETWORK socket1183 80 driver37c
SOCKET_NETWORK socket1184 80 driver37d
SOCKET_NETWORK socket1185 80 driver38a
SOCKET_NETWORK socket1186 80 driver38b
SOCKET_NETWORK socket1187 80 driver38c
SOCKET_NETWORK socket1188 80 driver38d
SOCKET_NETWORK socket1189 80 driver38a
SOCKET_NETWORK socket1190 80 driver38b
SOCKET_NETWORK socket1191 80 driver38c
SOCKET_NETWORK socket1192 80 driver38d
SOCKET_NETWORK socket1193 80 driver38a
SOCKET_NETWORK socket1194 80 driver38b
SOCKET_NETWORK socket1195 80 driver38c
SOCKET_NETWORK socket1196 80 driver38d
SOCKET_NETWORK socket1197 80 driver38a
SOCKET_NETWORK socket1198 80 driver38b
SOCKET_NETWORK socket1199 80 driver38c
SOCKET_NETWORK socket1200 80 driver38d
SOCKET_NETWORK socket1201 80 driver38a
SOCKET_NETWORK socket1202 80 driver38b
SOCKET_NETWORK socket1203 80 driver38c
SOCKET_NETWORK socket1204 80 driver38d
SOCKET_NETWORK socket1205 80 driver38a
SOCKET_NETWORK socket1206 80 driver38b
SOCKET_NETWORK socket1207 80 driver38c
SOCKET_NETWORK socket1208 80 driver38d
SOCKET_NETWORK socket1209 80 driver38a
SOCKET_NETWORK socket1210 80 driver38b
SOCKET_NETWORK socket1211 80 driver38c
SOCKET_NETWORK socket1212 80 driver38d
SOCKET_NETWORK socket1213 80 driver38a
SOCKET_NETWORK socket1214 80 driver38b
SOCKET_NETWORK socket1215 80 driver38c
SOCKET_NETWORK socket1216 80 driver38d
SOCKET_NETWORK socket1217 80 driver39a
SOCKET_NETWORK socket1218 80 driver39b
SOCKET_NETWORK socket1219 80 driver39c
SOCKET_NETWORK socket1220 80 driver39d
SOCKET_NETWORK socket1221 80 driver39a
SOCKET_NETWORK socket1222 80 driver39b
SOCKET_NETWORK socket1223 80 driver39c
SOCKET_NETWORK socket1224 80 driver39d
SOCKET_NETWORK socket1225 80 driver39a
SOCKET_NETWORK socket1226 80 driver39b
SOCKET_NETWORK socket1227 80 driver39c
SOCKET_NETWORK socket1228 80 driver39d
SOCKET_NETWORK socket1229 80 driver39a
SOCKET_NETWORK socket1230 80 driver39b
SOCKET_NETWORK socket1231 80 driver39c
SOCKET_NETWORK socket1232 80 driver39d
SOCKET_NETWORK socket1233 80 driver39a
SOCKET_NETWORK socket1234 80 driver39b

```

```

SOCKET_NETWORK socket1235 80 driver39c
SOCKET_NETWORK socket1236 80 driver39d
SOCKET_NETWORK socket1237 80 driver39a
SOCKET_NETWORK socket1238 80 driver39b
SOCKET_NETWORK socket1239 80 driver39c
SOCKET_NETWORK socket1240 80 driver39d
SOCKET_NETWORK socket1241 80 driver39a
SOCKET_NETWORK socket1242 80 driver39b
SOCKET_NETWORK socket1243 80 driver39c
SOCKET_NETWORK socket1244 80 driver39d
SOCKET_NETWORK socket1245 80 driver39a
SOCKET_NETWORK socket1246 80 driver39b
SOCKET_NETWORK socket1247 80 driver39c
SOCKET_NETWORK socket1248 80 driver39d
SOCKET_NETWORK socket1249 80 driver40a
SOCKET_NETWORK socket1250 80 driver40b
SOCKET_NETWORK socket1251 80 driver40c
SOCKET_NETWORK socket1252 80 driver40d
SOCKET_NETWORK socket1253 80 driver40a
SOCKET_NETWORK socket1254 80 driver40b
SOCKET_NETWORK socket1255 80 driver40c
SOCKET_NETWORK socket1256 80 driver40d
SOCKET_NETWORK socket1257 80 driver40a
SOCKET_NETWORK socket1258 80 driver40b
SOCKET_NETWORK socket1259 80 driver40c
SOCKET_NETWORK socket1260 80 driver40d
SOCKET_NETWORK socket1261 80 driver40a
SOCKET_NETWORK socket1262 80 driver40b
SOCKET_NETWORK socket1263 80 driver40c
SOCKET_NETWORK socket1264 80 driver40d
SOCKET_NETWORK socket1265 80 driver40a
SOCKET_NETWORK socket1266 80 driver40b
SOCKET_NETWORK socket1267 80 driver40c
SOCKET_NETWORK socket1268 80 driver40d
SOCKET_NETWORK socket1269 80 driver40a
SOCKET_NETWORK socket1270 80 driver40b
SOCKET_NETWORK socket1271 80 driver40c
SOCKET_NETWORK socket1272 80 driver40d
SOCKET_NETWORK socket1273 80 driver40a

SOCKET_NETWORK socket1274 80 driver40b
SOCKET_NETWORK socket1275 80 driver40c
SOCKET_NETWORK socket1276 80 driver40d
SOCKET_NETWORK socket1277 80 driver40a
SOCKET_NETWORK socket1278 80 driver40b
SOCKET_NETWORK socket1279 80 driver40c
SOCKET_NETWORK socket1280 80 driver40d
#endif
/*-----*/
OUTPUTNAME="regattaH"
CPU=48
#if 0
BEGIN_WAIT=5:00
RAMPUP=42:30
RUNTIME=30:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=17:00
#else
BEGIN_WAIT=25:00
RAMPUP=30:00
RUNTIME=150:00
RAMPDOWN_WAIT=3:00
RAMPDOWN=20:00
/*RAMPUP_SEC 1600 */
/*RUNTIME_SEC 9000 */
/*WAREHOUSES 81700 */
/*CLIENTS 1000 */
#endif
INTERVAL=1:00 /* Interval to calculate mix from */
LOGIN_MAX_LOAD = 8
LOGIN_BEGIN = 0 /* skip login state if set to 1 */
NOBEGIN = 1

```

```

KEYSTROKE_PACKET_SIZE = 0
MAX_CONCURRENT_SPAWN = 5
SPAWN_COUNT = 4
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 rteparams.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 630 0-63
START_RANGE client1b socket2 640 63-127
START_RANGE client1a socket3 640 127-191
START_RANGE client1b socket4 640 191-255
START_RANGE client1a socket5 640 255-319
START_RANGE client1b socket6 630 319-382
START_RANGE client1a socket7 640 382-446
START_RANGE client1b socket8 640 446-510
START_RANGE client1a socket9 640 510-574
START_RANGE client1b socket10 640 574-638
START_RANGE client1a socket11 640 638-702
START_RANGE client1b socket12 630 702-765
START_RANGE client1a socket13 640 765-829
START_RANGE client1b socket14 640 829-893
START_RANGE client1a socket15 640 893-957
START_RANGE client1b socket16 640 957-1021

START_RANGE client1a socket17 640 1021-1085
START_RANGE client1b socket18 630 1085-1148
START_RANGE client1a socket19 640 1148-1212
START_RANGE client1b socket20 640 1212-1276
START_RANGE client1a socket21 640 1276-1340
START_RANGE client1b socket22 640 1340-1404
START_RANGE client1a socket23 640 1404-1468
START_RANGE client1b socket24 630 1468-1531
START_RANGE client1a socket25 640 1531-1595
START_RANGE client1b socket26 640 1595-1659
START_RANGE client1a socket27 640 1659-1723
START_RANGE client1b socket28 640 1723-1787
START_RANGE client1a socket29 640 1787-1851
START_RANGE client1b socket30 630 1851-1914
START_RANGE client1a socket31 640 1914-1978
START_RANGE client1b socket32 640 1978-2042

START_RANGE client2a socket33 640 2042-2106
START_RANGE client2b socket34 640 2106-2170

START_RANGE client2a socket35 630 2170-2233
START_RANGE client2b socket36 640 2233-2297
START_RANGE client2a socket37 640 2297-2361
START_RANGE client2b socket38 640 2361-2425
START_RANGE client2a socket39 640 2425-2489
START_RANGE client2b socket40 640 2489-2553
START_RANGE client2a socket41 630 2553-2616

```

```

START_RANGE client2b socket42 640 2616-2680
START_RANGE client2a socket43 640 2680-2744
START_RANGE client2b socket44 640 2744-2808
START_RANGE client2a socket45 640 2808-2872
START_RANGE client2b socket46 640 2872-2936
START_RANGE client2a socket47 630 2936-2999
START_RANGE client2b socket48 640 2999-3063

```

```

START_RANGE client2a socket49 640 3063-3127
START_RANGE client2b socket50 640 3127-3191
START_RANGE client2a socket51 640 3191-3255
START_RANGE client2b socket52 640 3255-3319
START_RANGE client2a socket53 630 3319-3382
START_RANGE client2b socket54 640 3382-3446
START_RANGE client2a socket55 640 3446-3510
START_RANGE client2b socket56 640 3510-3574
START_RANGE client2a socket57 640 3574-3638
START_RANGE client2b socket58 640 3638-3702
START_RANGE client2a socket59 630 3702-3765
START_RANGE client2b socket60 640 3765-3829
START_RANGE client2a socket61 640 3829-3893
START_RANGE client2b socket62 640 3893-3957
START_RANGE client2a socket63 640 3957-4021
START_RANGE client2b socket64 640 4021-4085

```

```

START_RANGE client3a socket65 630 4085-4148
START_RANGE client3b socket66 640 4148-4212
START_RANGE client3a socket67 640 4212-4276
START_RANGE client3b socket68 640 4276-4340
START_RANGE client3a socket69 640 4340-4404
START_RANGE client3b socket70 630 4404-4467
START_RANGE client3a socket71 640 4467-4531
START_RANGE client3b socket72 640 4531-4595
START_RANGE client3a socket73 640 4595-4659
START_RANGE client3b socket74 640 4659-4723
START_RANGE client3a socket75 640 4723-4787
START_RANGE client3b socket76 630 4787-4850
START_RANGE client3a socket77 640 4850-4914
START_RANGE client3b socket78 640 4914-4978
START_RANGE client3a socket79 640 4978-5042
START_RANGE client3b socket80 640 5042-5106

```

```

START_RANGE client3a socket81 640 5106-5170
START_RANGE client3b socket82 630 5170-5233
START_RANGE client3a socket83 640 5233-5297
START_RANGE client3b socket84 640 5297-5361
START_RANGE client3a socket85 640 5361-5425
START_RANGE client3b socket86 640 5425-5489
START_RANGE client3a socket87 640 5489-5553
START_RANGE client3b socket88 630 5553-5616
START_RANGE client3a socket89 640 5616-5680
START_RANGE client3b socket90 640 5680-5744
START_RANGE client3a socket91 640 5744-5808
START_RANGE client3b socket92 640 5808-5872
START_RANGE client3a socket93 640 5872-5936
START_RANGE client3b socket94 630 5936-5999
START_RANGE client3a socket95 640 5999-6063
START_RANGE client3b socket96 640 6063-6127

```

```

START_RANGE client4a socket97 640 6127-6191
START_RANGE client4b socket98 640 6191-6255
START_RANGE client4a socket99 630 6255-6318
START_RANGE client4b socket100 640 6318-6382
START_RANGE client4a socket101 640 6382-6446
START_RANGE client4b socket102 640 6446-6510
START_RANGE client4a socket103 640 6510-6574
START_RANGE client4b socket104 640 6574-6638
START_RANGE client4a socket105 630 6638-6701
START_RANGE client4b socket106 640 6701-6765
START_RANGE client4a socket107 640 6765-6829
START_RANGE client4b socket108 640 6829-6893
START_RANGE client4a socket109 640 6893-6957

```

START_RANGE client4b socket110 640 6957-7021
START_RANGE client4a socket111 630 7021-7084
START_RANGE client4b socket112 640 7084-7148

START_RANGE client4a socket113 640 7148-7212
START_RANGE client4b socket114 640 7212-7276
START_RANGE client4a socket115 640 7276-7340
START_RANGE client4b socket116 640 7340-7404
START_RANGE client4a socket117 630 7404-7467
START_RANGE client4b socket118 640 7467-7531
START_RANGE client4a socket119 640 7531-7595
START_RANGE client4b socket120 640 7595-7659
START_RANGE client4a socket121 640 7659-7723
START_RANGE client4b socket122 640 7723-7787
START_RANGE client4a socket123 630 7787-7850
START_RANGE client4b socket124 640 7850-7914
START_RANGE client4a socket125 640 7914-7978
START_RANGE client4b socket126 640 7978-8042
START_RANGE client4a socket127 640 8042-8106
START_RANGE client4b socket128 640 8106-8170

#elif MASTER_NUM2

START_RANGE client5a socket129 630 8170-8233
START_RANGE client5b socket130 640 8233-8297
START_RANGE client5a socket131 640 8297-8361
START_RANGE client5b socket132 640 8361-8425
START_RANGE client5a socket133 640 8425-8489
START_RANGE client5b socket134 630 8489-8552
START_RANGE client5a socket135 640 8552-8616
START_RANGE client5b socket136 640 8616-8680
START_RANGE client5a socket137 640 8680-8744
START_RANGE client5b socket138 640 8744-8808
START_RANGE client5a socket139 640 8808-8872
START_RANGE client5b socket140 630 8872-8935
START_RANGE client5a socket141 640 8935-8999
START_RANGE client5b socket142 640 8999-9063
START_RANGE client5a socket143 640 9063-9127
START_RANGE client5b socket144 640 9127-9191

START_RANGE client5a socket145 640 9191-9255
START_RANGE client5b socket146 630 9255-9318

START_RANGE client5a socket147 640 9318-9382
START_RANGE client5b socket148 640 9382-9446
START_RANGE client5a socket149 640 9446-9510
START_RANGE client5b socket150 640 9510-9574
START_RANGE client5a socket151 640 9574-9638
START_RANGE client5b socket152 630 9638-9701
START_RANGE client5a socket153 640 9701-9765
START_RANGE client5b socket154 640 9765-9829
START_RANGE client5a socket155 640 9829-9893
START_RANGE client5b socket156 640 9893-9957
START_RANGE client5a socket157 640 9957-10021
START_RANGE client5b socket158 630 10021-10084
START_RANGE client5a socket159 640 10084-10148
START_RANGE client5b socket160 640 10148-10212

START_RANGE client6a socket161 640 10212-10276
START_RANGE client6b socket162 640 10276-10340
START_RANGE client6a socket163 630 10340-10403
START_RANGE client6b socket164 640 10403-10467
START_RANGE client6a socket165 640 10467-10531
START_RANGE client6b socket166 640 10531-10595
START_RANGE client6a socket167 640 10595-10659
START_RANGE client6b socket168 640 10659-10723
START_RANGE client6a socket169 630 10723-10786
START_RANGE client6b socket170 640 10786-10850
START_RANGE client6a socket171 640 10850-10914
START_RANGE client6b socket172 640 10914-10978
START_RANGE client6a socket173 640 10978-11042
START_RANGE client6b socket174 640 11042-11106
START_RANGE client6a socket175 630 11106-11169

START_RANGE client6b socket176 640 11169-11233

START_RANGE client6a socket177 640 11233-11297
START_RANGE client6b socket178 640 11297-11361
START_RANGE client6a socket179 640 11361-11425
START_RANGE client6b socket180 640 11425-11489
START_RANGE client6a socket181 630 11489-11552
START_RANGE client6b socket182 640 11552-11616
START_RANGE client6a socket183 640 11616-11680
START_RANGE client6b socket184 640 11680-11744
START_RANGE client6a socket185 640 11744-11808
START_RANGE client6b socket186 640 11808-11872
START_RANGE client6a socket187 630 11872-11935
START_RANGE client6b socket188 640 11935-11999
START_RANGE client6a socket189 640 11999-12063
START_RANGE client6b socket190 640 12063-12127
START_RANGE client6a socket191 640 12127-12191
START_RANGE client6b socket192 640 12191-12255

START_RANGE client7a socket193 630 12255-12318
START_RANGE client7b socket194 640 12318-12382
START_RANGE client7a socket195 640 12382-12446
START_RANGE client7b socket196 640 12446-12510
START_RANGE client7a socket197 640 12510-12574
START_RANGE client7b socket198 630 12574-12637
START_RANGE client7a socket199 640 12637-12701
START_RANGE client7b socket200 640 12701-12765
START_RANGE client7a socket201 640 12765-12829
START_RANGE client7b socket202 640 12829-12893
START_RANGE client7a socket203 640 12893-12957
START_RANGE client7b socket204 630 12957-13020
START_RANGE client7a socket205 640 13020-13084
START_RANGE client7b socket206 640 13084-13148
START_RANGE client7a socket207 640 13148-13212
START_RANGE client7b socket208 640 13212-13276

START_RANGE client7a socket209 640 13276-13340
START_RANGE client7b socket210 630 13340-13403
START_RANGE client7a socket211 640 13403-13467
START_RANGE client7b socket212 640 13467-13531
START_RANGE client7a socket213 640 13531-13595
START_RANGE client7b socket214 640 13595-13659
START_RANGE client7a socket215 640 13659-13723
START_RANGE client7b socket216 630 13723-13786
START_RANGE client7a socket217 640 13786-13850
START_RANGE client7b socket218 640 13850-13914
START_RANGE client7a socket219 640 13914-13978
START_RANGE client7b socket220 640 13978-14042
START_RANGE client7a socket221 640 14042-14106
START_RANGE client7b socket222 630 14106-14169
START_RANGE client7a socket223 640 14169-14233
START_RANGE client7b socket224 640 14233-14297

START_RANGE client8a socket225 640 14297-14361
START_RANGE client8b socket226 640 14361-14425
START_RANGE client8a socket227 630 14425-14488
START_RANGE client8b socket228 640 14488-14552
START_RANGE client8a socket229 640 14552-14616
START_RANGE client8b socket230 640 14616-14680
START_RANGE client8a socket231 640 14680-14744
START_RANGE client8b socket232 640 14744-14808
START_RANGE client8a socket233 630 14808-14871
START_RANGE client8b socket234 640 14871-14935
START_RANGE client8a socket235 640 14935-14999
START_RANGE client8b socket236 640 14999-15063
START_RANGE client8a socket237 640 15063-15127
START_RANGE client8b socket238 640 15127-15191
START_RANGE client8a socket239 630 15191-15254
START_RANGE client8b socket240 640 15254-15318

START_RANGE client8a socket241 640 15318-15382
START_RANGE client8b socket242 640 15382-15446

START_RANGE client8a socket243 640 15446-15510
START_RANGE client8b socket244 640 15510-15574
START_RANGE client8a socket245 630 15574-15637
START_RANGE client8b socket246 640 15637-15701
START_RANGE client8a socket247 640 15701-15765
START_RANGE client8b socket248 640 15765-15829
START_RANGE client8a socket249 640 15829-15893
START_RANGE client8b socket250 640 15893-15957
START_RANGE client8a socket251 630 15957-16020
START_RANGE client8b socket252 640 16020-16084
START_RANGE client8a socket253 640 16084-16148
START_RANGE client8b socket254 640 16148-16212
START_RANGE client8a socket255 640 16212-16276
START_RANGE client8b socket256 640 16276-16340

#elif MASTER_NUM3

START_RANGE client9a socket257 630 16340-16403
START_RANGE client9b socket258 640 16403-16467
START_RANGE client9a socket259 640 16467-16531
START_RANGE client9b socket260 640 16531-16595
START_RANGE client9a socket261 640 16595-16659
START_RANGE client9b socket262 630 16659-16722
START_RANGE client9a socket263 640 16722-16786
START_RANGE client9b socket264 640 16786-16850
START_RANGE client9a socket265 640 16850-16914
START_RANGE client9b socket266 640 16914-16978
START_RANGE client9a socket267 640 16978-17042
START_RANGE client9b socket268 630 17042-17105
START_RANGE client9a socket269 640 17105-17169
START_RANGE client9b socket270 640 17169-17233
START_RANGE client9a socket271 640 17233-17297
START_RANGE client9b socket272 640 17297-17361

START_RANGE client9a socket273 640 17361-17425
START_RANGE client9b socket274 630 17425-17488
START_RANGE client9a socket275 640 17488-17552
START_RANGE client9b socket276 640 17552-17616
START_RANGE client9a socket277 640 17616-17680
START_RANGE client9b socket278 640 17680-17744
START_RANGE client9a socket279 640 17744-17808
START_RANGE client9b socket280 630 17808-17871
START_RANGE client9a socket281 640 17871-17935
START_RANGE client9b socket282 640 17935-17999
START_RANGE client9a socket283 640 17999-18063
START_RANGE client9b socket284 640 18063-18127
START_RANGE client9a socket285 640 18127-18191
START_RANGE client9b socket286 630 18191-18254
START_RANGE client9a socket287 640 18254-18318
START_RANGE client9b socket288 640 18318-18382

START_RANGE client10a socket289 640 18382-18446
START_RANGE client10b socket290 640 18446-18510
START_RANGE client10a socket291 630 18510-18573
START_RANGE client10b socket292 640 18573-18637
START_RANGE client10a socket293 640 18637-18701
START_RANGE client10b socket294 640 18701-18765
START_RANGE client10a socket295 640 18765-18829
START_RANGE client10b socket296 640 18829-18893
START_RANGE client10a socket297 630 18893-18956
START_RANGE client10b socket298 640 18956-19020
START_RANGE client10a socket299 640 19020-19084
START_RANGE client10b socket300 640 19084-19148
START_RANGE client10a socket301 640 19148-19212
START_RANGE client10b socket302 640 19212-19276
START_RANGE client10a socket303 630 19276-19339
START_RANGE client10b socket304 640 19339-19403

START_RANGE client10a socket305 640 19403-19467
START_RANGE client10b socket306 640 19467-19531
START_RANGE client10a socket307 640 19531-19595
START_RANGE client10b socket308 640 19595-19659
START_RANGE client10a socket309 630 19659-19722

START_RANGE client10b socket310 640 19722-19786
START_RANGE client10a socket311 640 19786-19850
START_RANGE client10b socket312 640 19850-19914
START_RANGE client10a socket313 640 19914-19978
START_RANGE client10b socket314 640 19978-20042
START_RANGE client10a socket315 630 20042-20105
START_RANGE client10b socket316 640 20105-20169
START_RANGE client10a socket317 640 20169-20233
START_RANGE client10b socket318 640 20233-20297
START_RANGE client10a socket319 640 20297-20361
START_RANGE client10b socket320 640 20361-20425

START_RANGE client11a socket321 630 20425-20488
START_RANGE client11b socket322 640 20488-20552
START_RANGE client11a socket323 640 20552-20616
START_RANGE client11b socket324 640 20616-20680
START_RANGE client11a socket325 640 20680-20744
START_RANGE client11b socket326 630 20744-20807
START_RANGE client11a socket327 640 20807-20871
START_RANGE client11b socket328 640 20871-20935
START_RANGE client11a socket329 640 20935-20999
START_RANGE client11b socket330 640 20999-21063
START_RANGE client11a socket331 640 21063-21127
START_RANGE client11b socket332 630 21127-21190
START_RANGE client11a socket333 640 21190-21254
START_RANGE client11b socket334 640 21254-21318
START_RANGE client11a socket335 640 21318-21382
START_RANGE client11b socket336 640 21382-21446

START_RANGE client11a socket337 640 21446-21510
START_RANGE client11b socket338 630 21510-21573
START_RANGE client11a socket339 640 21573-21637
START_RANGE client11b socket340 640 21637-21701
START_RANGE client11a socket341 640 21701-21765
START_RANGE client11b socket342 640 21765-21829
START_RANGE client11a socket343 640 21829-21893
START_RANGE client11b socket344 630 21893-21956
START_RANGE client11a socket345 640 21956-22020
START_RANGE client11b socket346 640 22020-22084
START_RANGE client11a socket347 640 22084-22148
START_RANGE client11b socket348 640 22148-22212
START_RANGE client11a socket349 640 22212-22276
START_RANGE client11b socket350 630 22276-22339
START_RANGE client11a socket351 640 22339-22403
START_RANGE client11b socket352 640 22403-22467

START_RANGE client12a socket353 640 22467-22531
START_RANGE client12b socket354 640 22531-22595
START_RANGE client12a socket355 630 22595-22658
START_RANGE client12b socket356 640 22658-22722
START_RANGE client12a socket357 640 22722-22786
START_RANGE client12b socket358 640 22786-22850
START_RANGE client12a socket359 640 22850-22914
START_RANGE client12b socket360 640 22914-22978
START_RANGE client12a socket361 630 22978-23041
START_RANGE client12b socket362 640 23041-23105
START_RANGE client12a socket363 640 23105-23169
START_RANGE client12b socket364 640 23169-23233
START_RANGE client12a socket365 640 23233-23297
START_RANGE client12b socket366 640 23297-23361
START_RANGE client12a socket367 630 23361-23424
START_RANGE client12b socket368 640 23424-23488

START_RANGE client12a socket369 640 23488-23552
START_RANGE client12b socket370 640 23552-23616
START_RANGE client12a socket371 640 23616-23680
START_RANGE client12b socket372 640 23680-23744
START_RANGE client12a socket373 630 23744-23807
START_RANGE client12b socket374 640 23807-23871
START_RANGE client12a socket375 640 23871-23935
START_RANGE client12b socket376 640 23935-23999
START_RANGE client12a socket377 640 23999-24063

START_RANGE client12b socket378 640 24063-24127
START_RANGE client12a socket379 630 24127-24190
START_RANGE client12b socket380 640 24190-24254
START_RANGE client12a socket381 640 24254-24318
START_RANGE client12b socket382 640 24318-24382
START_RANGE client12a socket383 640 24382-24446
START_RANGE client12b socket384 640 24446-24510

#elif MASTER_NUM4

START_RANGE client13a socket385 630 24510-24573
START_RANGE client13b socket386 640 24573-24637
START_RANGE client13a socket387 640 24637-24701
START_RANGE client13b socket388 640 24701-24765
START_RANGE client13a socket389 640 24765-24829
START_RANGE client13b socket390 630 24829-24892
START_RANGE client13a socket391 640 24892-24956
START_RANGE client13b socket392 640 24956-25020
START_RANGE client13a socket393 640 25020-25084
START_RANGE client13b socket394 640 25084-25148
START_RANGE client13a socket395 640 25148-25212
START_RANGE client13b socket396 630 25212-25275
START_RANGE client13a socket397 640 25275-25339
START_RANGE client13b socket398 640 25339-25403
START_RANGE client13a socket399 640 25403-25467
START_RANGE client13b socket400 640 25467-25531

START_RANGE client13a socket401 640 25531-25595
START_RANGE client13b socket402 630 25595-25658
START_RANGE client13a socket403 640 25658-25722
START_RANGE client13b socket404 640 25722-25786
START_RANGE client13a socket405 640 25786-25850
START_RANGE client13b socket406 640 25850-25914
START_RANGE client13a socket407 640 25914-25978
START_RANGE client13b socket408 630 25978-26041
START_RANGE client13a socket409 640 26041-26105
START_RANGE client13b socket410 640 26105-26169
START_RANGE client13a socket411 640 26169-26233
START_RANGE client13b socket412 640 26233-26297
START_RANGE client13a socket413 640 26297-26361
START_RANGE client13b socket414 630 26361-26424
START_RANGE client13a socket415 640 26424-26488
START_RANGE client13b socket416 640 26488-26552

START_RANGE client14a socket417 640 26552-26616
START_RANGE client14b socket418 640 26616-26680
START_RANGE client14a socket419 630 26680-26743
START_RANGE client14b socket420 640 26743-26807
START_RANGE client14a socket421 640 26807-26871
START_RANGE client14b socket422 640 26871-26935
START_RANGE client14a socket423 640 26935-26999
START_RANGE client14b socket424 640 26999-27063
START_RANGE client14a socket425 630 27063-27126
START_RANGE client14b socket426 640 27126-27190
START_RANGE client14a socket427 640 27190-27254
START_RANGE client14b socket428 640 27254-27318
START_RANGE client14a socket429 640 27318-27382
START_RANGE client14b socket430 640 27382-27446
START_RANGE client14a socket431 630 27446-27509
START_RANGE client14b socket432 640 27509-27573

START_RANGE client14a socket433 640 27573-27637
START_RANGE client14b socket434 640 27637-27701
START_RANGE client14a socket435 640 27701-27765
START_RANGE client14b socket436 640 27765-27829
START_RANGE client14a socket437 630 27829-27892
START_RANGE client14b socket438 640 27892-27956
START_RANGE client14a socket439 640 27956-28020
START_RANGE client14b socket440 640 28020-28084
START_RANGE client14a socket441 640 28084-28148
START_RANGE client14b socket442 640 28148-28212
START_RANGE client14a socket443 630 28212-28275
START_RANGE client14b socket444 640 28275-28339

START_RANGE client14a socket445 640 28339-28403
START_RANGE client14b socket446 640 28403-28467
START_RANGE client14a socket447 640 28467-28531
START_RANGE client14b socket448 640 28531-28595

START_RANGE client15a socket449 630 28595-28658
START_RANGE client15b socket450 640 28658-28722
START_RANGE client15a socket451 640 28722-28786
START_RANGE client15b socket452 640 28786-28850
START_RANGE client15a socket453 640 28850-28914
START_RANGE client15b socket454 630 28914-28977
START_RANGE client15a socket455 640 28977-29041
START_RANGE client15b socket456 640 29041-29105
START_RANGE client15a socket457 640 29105-29169
START_RANGE client15b socket458 640 29169-29233
START_RANGE client15a socket459 640 29233-29297
START_RANGE client15b socket460 630 29297-29360
START_RANGE client15a socket461 640 29360-29424
START_RANGE client15b socket462 640 29424-29488
START_RANGE client15a socket463 640 29488-29552
START_RANGE client15b socket464 640 29552-29616

START_RANGE client15a socket465 640 29616-29680
START_RANGE client15b socket466 630 29680-29743
START_RANGE client15a socket467 640 29743-29807
START_RANGE client15b socket468 640 29807-29871
START_RANGE client15a socket469 640 29871-29935
START_RANGE client15b socket470 640 29935-29999
START_RANGE client15a socket471 640 29999-30063
START_RANGE client15b socket472 630 30063-30126
START_RANGE client15a socket473 640 30126-30190
START_RANGE client15b socket474 640 30190-30254
START_RANGE client15a socket475 640 30254-30318
START_RANGE client15b socket476 640 30318-30382
START_RANGE client15a socket477 640 30382-30446
START_RANGE client15b socket478 630 30446-30509
START_RANGE client15a socket479 640 30509-30573
START_RANGE client15b socket480 640 30573-30637

START_RANGE client16a socket481 640 30637-30701
START_RANGE client16b socket482 640 30701-30765
START_RANGE client16a socket483 630 30765-30828
START_RANGE client16b socket484 640 30828-30892
START_RANGE client16a socket485 640 30892-30956
START_RANGE client16b socket486 640 30956-31020
START_RANGE client16a socket487 640 31020-31084
START_RANGE client16b socket488 640 31084-31148
START_RANGE client16a socket489 630 31148-31211
START_RANGE client16b socket490 640 31211-31275
START_RANGE client16a socket491 640 31275-31339
START_RANGE client16b socket492 640 31339-31403
START_RANGE client16a socket493 640 31403-31467
START_RANGE client16b socket494 640 31467-31531
START_RANGE client16a socket495 630 31531-31594
START_RANGE client16b socket496 640 31594-31658

START_RANGE client16a socket497 640 31658-31722
START_RANGE client16b socket498 640 31722-31786
START_RANGE client16a socket499 640 31786-31850
START_RANGE client16b socket500 640 31850-31914
START_RANGE client16a socket501 630 31914-31977
START_RANGE client16b socket502 640 31977-32041
START_RANGE client16a socket503 640 32041-32105
START_RANGE client16b socket504 640 32105-32169
START_RANGE client16a socket505 640 32169-32233
START_RANGE client16b socket506 640 32233-32297
START_RANGE client16a socket507 630 32297-32360
START_RANGE client16b socket508 640 32360-32424
START_RANGE client16a socket509 640 32424-32488
START_RANGE client16b socket510 640 32488-32552
START_RANGE client16a socket511 640 32552-32616
START_RANGE client16b socket512 640 32616-32680

#elif MASTER_NUM5

START_RANGE client17a socket513 630 32680-32743
START_RANGE client17b socket514 640 32743-32807
START_RANGE client17a socket515 640 32807-32871
START_RANGE client17b socket516 640 32871-32935
START_RANGE client17a socket517 640 32935-32999
START_RANGE client17b socket518 630 32999-33062
START_RANGE client17a socket519 640 33062-33126
START_RANGE client17b socket520 640 33126-33190
START_RANGE client17a socket521 640 33190-33254
START_RANGE client17b socket522 640 33254-33318
START_RANGE client17a socket523 640 33318-33382
START_RANGE client17b socket524 630 33382-33445
START_RANGE client17a socket525 640 33445-33509
START_RANGE client17b socket526 640 33509-33573
START_RANGE client17a socket527 640 33573-33637
START_RANGE client17b socket528 640 33637-33701

START_RANGE client17a socket529 640 33701-33765
START_RANGE client17b socket530 630 33765-33828
START_RANGE client17a socket531 640 33828-33892
START_RANGE client17b socket532 640 33892-33956
START_RANGE client17a socket533 640 33956-34020
START_RANGE client17b socket534 640 34020-34084
START_RANGE client17a socket535 640 34084-34148
START_RANGE client17b socket536 630 34148-34211
START_RANGE client17a socket537 640 34211-34275
START_RANGE client17b socket538 640 34275-34339
START_RANGE client17a socket539 640 34339-34403
START_RANGE client17b socket540 640 34403-34467
START_RANGE client17a socket541 640 34467-34531
START_RANGE client17b socket542 630 34531-34594
START_RANGE client17a socket543 640 34594-34658
START_RANGE client17b socket544 640 34658-34722

START_RANGE client18a socket545 640 34722-34786
START_RANGE client18b socket546 640 34786-34850
START_RANGE client18a socket547 630 34850-34913
START_RANGE client18b socket548 640 34913-34977
START_RANGE client18a socket549 640 34977-35041
START_RANGE client18b socket550 640 35041-35105
START_RANGE client18a socket551 640 35105-35169
START_RANGE client18b socket552 640 35169-35233
START_RANGE client18a socket553 630 35233-35296
START_RANGE client18b socket554 640 35296-35360
START_RANGE client18a socket555 640 35360-35424
START_RANGE client18b socket556 640 35424-35488
START_RANGE client18a socket557 640 35488-35552
START_RANGE client18b socket558 640 35552-35616
START_RANGE client18a socket559 630 35616-35679
START_RANGE client18b socket560 640 35679-35743

START_RANGE client18a socket561 640 35743-35807
START_RANGE client18b socket562 640 35807-35871
START_RANGE client18a socket563 640 35871-35935
START_RANGE client18b socket564 640 35935-35999
START_RANGE client18a socket565 630 35999-36062
START_RANGE client18b socket566 640 36062-36126
START_RANGE client18a socket567 640 36126-36190
START_RANGE client18b socket568 640 36190-36254
START_RANGE client18a socket569 640 36254-36318
START_RANGE client18b socket570 640 36318-36382
START_RANGE client18a socket571 630 36382-36445
START_RANGE client18b socket572 640 36445-36509
START_RANGE client18a socket573 640 36509-36573
START_RANGE client18b socket574 640 36573-36637
START_RANGE client18a socket575 640 36637-36701
START_RANGE client18b socket576 640 36701-36765

START_RANGE client19a socket577 630 36765-36828
START_RANGE client19b socket578 640 36828-36892

START_RANGE client19a socket579 640 36892-36956
START_RANGE client19b socket580 640 36956-37020
START_RANGE client19a socket581 640 37020-37084
START_RANGE client19b socket582 630 37084-37147
START_RANGE client19a socket583 640 37147-37211
START_RANGE client19b socket584 640 37211-37275
START_RANGE client19a socket585 640 37275-37339
START_RANGE client19b socket586 640 37339-37403
START_RANGE client19a socket587 640 37403-37467
START_RANGE client19b socket588 630 37467-37530
START_RANGE client19a socket589 640 37530-37594
START_RANGE client19b socket590 640 37594-37658
START_RANGE client19a socket591 640 37658-37722
START_RANGE client19b socket592 640 37722-37786

START_RANGE client19a socket593 640 37786-37850
START_RANGE client19b socket594 630 37850-37913
START_RANGE client19a socket595 640 37913-37977
START_RANGE client19b socket596 640 37977-38041
START_RANGE client19a socket597 640 38041-38105
START_RANGE client19b socket598 640 38105-38169
START_RANGE client19a socket599 640 38169-38233
START_RANGE client19b socket600 630 38233-38296
START_RANGE client19a socket601 640 38296-38360
START_RANGE client19b socket602 640 38360-38424
START_RANGE client19a socket603 640 38424-38488
START_RANGE client19b socket604 640 38488-38552
START_RANGE client19a socket605 640 38552-38616
START_RANGE client19b socket606 630 38616-38679
START_RANGE client19a socket607 640 38679-38743
START_RANGE client19b socket608 640 38743-38807

START_RANGE client20a socket609 640 38807-38871
START_RANGE client20b socket610 640 38871-38935
START_RANGE client20a socket611 630 38935-38998
START_RANGE client20b socket612 640 38998-39062
START_RANGE client20a socket613 640 39062-39126
START_RANGE client20b socket614 640 39126-39190
START_RANGE client20a socket615 640 39190-39254
START_RANGE client20b socket616 640 39254-39318
START_RANGE client20a socket617 630 39318-39381
START_RANGE client20b socket618 640 39381-39445
START_RANGE client20a socket619 640 39445-39509
START_RANGE client20b socket620 640 39509-39573
START_RANGE client20a socket621 640 39573-39637
START_RANGE client20b socket622 640 39637-39701
START_RANGE client20a socket623 630 39701-39764
START_RANGE client20b socket624 640 39764-39828

START_RANGE client20a socket625 640 39828-39892
START_RANGE client20b socket626 640 39892-39956
START_RANGE client20a socket627 640 39956-40020
START_RANGE client20b socket628 640 40020-40084
START_RANGE client20a socket629 630 40084-40147
START_RANGE client20b socket630 640 40147-40211
START_RANGE client20a socket631 640 40211-40275
START_RANGE client20b socket632 640 40275-40339
START_RANGE client20a socket633 640 40339-40403
START_RANGE client20b socket634 640 40403-40467
START_RANGE client20a socket635 630 40467-40530
START_RANGE client20b socket636 640 40530-40594
START_RANGE client20a socket637 640 40594-40658
START_RANGE client20b socket638 640 40658-40722
START_RANGE client20a socket639 640 40722-40786
START_RANGE client20b socket640 640 40786-40850

#elif MASTER_NUM6

START_RANGE client21a socket641 630 40850-40913
START_RANGE client21b socket642 640 40913-40977
START_RANGE client21a socket643 640 40977-41041
START_RANGE client21b socket644 640 41041-41105
START_RANGE client21a socket645 640 41105-41169

START_RANGE client21b socket646 630 41169-41232
START_RANGE client21a socket647 640 41232-41296
START_RANGE client21b socket648 640 41296-41360
START_RANGE client21a socket649 640 41360-41424
START_RANGE client21b socket650 640 41424-41488
START_RANGE client21a socket651 640 41488-41552
START_RANGE client21b socket652 630 41552-41615
START_RANGE client21a socket653 640 41615-41679
START_RANGE client21b socket654 640 41679-41743
START_RANGE client21a socket655 640 41743-41807
START_RANGE client21b socket656 640 41807-41871

START_RANGE client21a socket657 640 41871-41935
START_RANGE client21b socket658 630 41935-41998
START_RANGE client21a socket659 640 41998-42062
START_RANGE client21b socket660 640 42062-42126
START_RANGE client21a socket661 640 42126-42190
START_RANGE client21b socket662 640 42190-42254
START_RANGE client21a socket663 640 42254-42318
START_RANGE client21b socket664 630 42318-42381
START_RANGE client21a socket665 640 42381-42445
START_RANGE client21b socket666 640 42445-42509
START_RANGE client21a socket667 640 42509-42573
START_RANGE client21b socket668 640 42573-42637
START_RANGE client21a socket669 640 42637-42701
START_RANGE client21b socket670 630 42701-42764
START_RANGE client21a socket671 640 42764-42828
START_RANGE client21b socket672 640 42828-42892

START_RANGE client22a socket673 640 42892-42956
START_RANGE client22b socket674 640 42956-43020
START_RANGE client22a socket675 630 43020-43083
START_RANGE client22b socket676 640 43083-43147
START_RANGE client22a socket677 640 43147-43211
START_RANGE client22b socket678 640 43211-43275
START_RANGE client22a socket679 640 43275-43339
START_RANGE client22b socket680 640 43339-43403
START_RANGE client22a socket681 630 43403-43466
START_RANGE client22b socket682 640 43466-43530
START_RANGE client22a socket683 640 43530-43594
START_RANGE client22b socket684 640 43594-43658
START_RANGE client22a socket685 640 43658-43722
START_RANGE client22b socket686 640 43722-43786
START_RANGE client22a socket687 630 43786-43849
START_RANGE client22b socket688 640 43849-43913

START_RANGE client22a socket689 640 43913-43977
START_RANGE client22b socket690 640 43977-44041
START_RANGE client22a socket691 640 44041-44105
START_RANGE client22b socket692 640 44105-44169
START_RANGE client22a socket693 630 44169-44232
START_RANGE client22b socket694 640 44232-44296
START_RANGE client22a socket695 640 44296-44360
START_RANGE client22b socket696 640 44360-44424
START_RANGE client22a socket697 640 44424-44488
START_RANGE client22b socket698 640 44488-44552
START_RANGE client22a socket699 630 44552-44615
START_RANGE client22b socket700 640 44615-44679
START_RANGE client22a socket701 640 44679-44743
START_RANGE client22b socket702 640 44743-44807
START_RANGE client22a socket703 640 44807-44871
START_RANGE client22b socket704 640 44871-44935

START_RANGE client23a socket705 630 44935-44998
START_RANGE client23b socket706 640 44998-45062
START_RANGE client23a socket707 640 45062-45126
START_RANGE client23b socket708 640 45126-45190
START_RANGE client23a socket709 640 45190-45254
START_RANGE client23b socket710 630 45254-45317
START_RANGE client23a socket711 640 45317-45381
START_RANGE client23b socket712 640 45381-45445
START_RANGE client23a socket713 640 45445-45509

START_RANGE client23b socket714 640 45509-45573
START_RANGE client23a socket715 640 45573-45637
START_RANGE client23b socket716 630 45637-45700
START_RANGE client23a socket717 640 45700-45764
START_RANGE client23b socket718 640 45764-45828
START_RANGE client23a socket719 640 45828-45892
START_RANGE client23b socket720 640 45892-45956

START_RANGE client23a socket721 640 45956-46020

START_RANGE client23b socket722 630 46020-46083
START_RANGE client23a socket723 640 46083-46147
START_RANGE client23b socket724 640 46147-46211
START_RANGE client23a socket725 640 46211-46275
START_RANGE client23b socket726 640 46275-46339
START_RANGE client23a socket727 640 46339-46403
START_RANGE client23b socket728 630 46403-46466
START_RANGE client23a socket729 640 46466-46530
START_RANGE client23b socket730 640 46530-46594
START_RANGE client23a socket731 640 46594-46658
START_RANGE client23b socket732 640 46658-46722
START_RANGE client23a socket733 640 46722-46786
START_RANGE client23b socket734 630 46786-46849
START_RANGE client23a socket735 640 46849-46913
START_RANGE client23b socket736 640 46913-46977

START_RANGE client24a socket737 640 46977-47041
START_RANGE client24b socket738 640 47041-47105
START_RANGE client24a socket739 630 47105-47168
START_RANGE client24b socket740 640 47168-47232
START_RANGE client24a socket741 640 47232-47296
START_RANGE client24b socket742 640 47296-47360
START_RANGE client24a socket743 640 47360-47424
START_RANGE client24b socket744 640 47424-47488
START_RANGE client24a socket745 630 47488-47551
START_RANGE client24b socket746 640 47551-47615
START_RANGE client24a socket747 640 47615-47679
START_RANGE client24b socket748 640 47679-47743
START_RANGE client24a socket749 640 47743-47807
START_RANGE client24b socket750 640 47807-47871
START_RANGE client24a socket751 630 47871-47934
START_RANGE client24b socket752 640 47934-47998

START_RANGE client24a socket753 640 47998-48062
START_RANGE client24b socket754 640 48062-48126
START_RANGE client24a socket755 640 48126-48190
START_RANGE client24b socket756 640 48190-48254
START_RANGE client24a socket757 630 48254-48317
START_RANGE client24b socket758 640 48317-48381
START_RANGE client24a socket759 640 48381-48445
START_RANGE client24b socket760 640 48445-48509
START_RANGE client24a socket761 640 48509-48573
START_RANGE client24b socket762 640 48573-48637
START_RANGE client24a socket763 630 48637-48700
START_RANGE client24b socket764 640 48700-48764
START_RANGE client24a socket765 640 48764-48828
START_RANGE client24b socket766 640 48828-48892
START_RANGE client24a socket767 640 48892-48956
START_RANGE client24b socket768 640 48956-49020

#elif MASTER_NUM7

START_RANGE client25a socket769 630 49020-49083
START_RANGE client25b socket770 640 49083-49147
START_RANGE client25a socket771 640 49147-49211
START_RANGE client25b socket772 640 49211-49275
START_RANGE client25a socket773 640 49275-49339
START_RANGE client25b socket774 630 49339-49402
START_RANGE client25a socket775 640 49402-49466
START_RANGE client25b socket776 640 49466-49530
START_RANGE client25a socket777 640 49530-49594
START_RANGE client25b socket778 640 49594-49658
START_RANGE client25a socket779 640 49658-49722

START_RANGE client25b socket780 630 49722-49785
START_RANGE client25a socket781 640 49785-49849
START_RANGE client25b socket782 640 49849-49913
START_RANGE client25a socket783 640 49913-49977
START_RANGE client25b socket784 640 49977-50041

START_RANGE client25a socket785 640 50041-50105
START_RANGE client25b socket786 630 50105-50168
START_RANGE client25a socket787 640 50168-50232
START_RANGE client25b socket788 640 50232-50296
START_RANGE client25a socket789 640 50296-50360
START_RANGE client25b socket790 640 50360-50424
START_RANGE client25a socket791 640 50424-50488
START_RANGE client25b socket792 630 50488-50551
START_RANGE client25a socket793 640 50551-50615
START_RANGE client25b socket794 640 50615-50679
START_RANGE client25a socket795 640 50679-50743
START_RANGE client25b socket796 640 50743-50807
START_RANGE client25a socket797 640 50807-50871
START_RANGE client25b socket798 630 50871-50934
START_RANGE client25a socket799 640 50934-50998
START_RANGE client25b socket800 640 50998-51062

START_RANGE client26a socket801 640 51062-51126
START_RANGE client26b socket802 640 51126-51190
START_RANGE client26a socket803 630 51190-51253
START_RANGE client26b socket804 640 51253-51317

START_RANGE client26a socket805 640 51317-51381
START_RANGE client26b socket806 640 51381-51445
START_RANGE client26a socket807 640 51445-51509
START_RANGE client26b socket808 640 51509-51573
START_RANGE client26a socket809 630 51573-51636
START_RANGE client26b socket810 640 51636-51700
START_RANGE client26a socket811 640 51700-51764
START_RANGE client26b socket812 640 51764-51828
START_RANGE client26a socket813 640 51828-51892
START_RANGE client26b socket814 640 51892-51956
START_RANGE client26a socket815 630 51956-52019
START_RANGE client26b socket816 640 52019-52083

START_RANGE client26a socket817 640 52083-52147
START_RANGE client26b socket818 640 52147-52211
START_RANGE client26a socket819 640 52211-52275
START_RANGE client26b socket820 640 52275-52339
START_RANGE client26a socket821 630 52339-52402
START_RANGE client26b socket822 640 52402-52466
START_RANGE client26a socket823 640 52466-52530
START_RANGE client26b socket824 640 52530-52594
START_RANGE client26a socket825 640 52594-52658
START_RANGE client26b socket826 640 52658-52722
START_RANGE client26a socket827 630 52722-52785
START_RANGE client26b socket828 640 52785-52849
START_RANGE client26a socket829 640 52849-52913
START_RANGE client26b socket830 640 52913-52977
START_RANGE client26a socket831 640 52977-53041
START_RANGE client26b socket832 640 53041-53105

START_RANGE client27a socket833 630 53105-53168
START_RANGE client27b socket834 640 53168-53232
START_RANGE client27a socket835 640 53232-53296
START_RANGE client27b socket836 640 53296-53360
START_RANGE client27a socket837 640 53360-53424
START_RANGE client27b socket838 630 53424-53487
START_RANGE client27a socket839 640 53487-53551
START_RANGE client27b socket840 640 53551-53615
START_RANGE client27a socket841 640 53615-53679
START_RANGE client27b socket842 640 53679-53743
START_RANGE client27a socket843 640 53743-53807
START_RANGE client27b socket844 630 53807-53870
START_RANGE client27a socket845 640 53870-53934
START_RANGE client27b socket846 640 53934-53998

START_RANGE client27a socket847 640 53998-54062
START_RANGE client27b socket848 640 54062-54126

START_RANGE client27a socket849 640 54126-54190
START_RANGE client27b socket850 630 54190-54253
START_RANGE client27a socket851 640 54253-54317
START_RANGE client27b socket852 640 54317-54381
START_RANGE client27a socket853 640 54381-54445
START_RANGE client27b socket854 640 54445-54509
START_RANGE client27a socket855 640 54509-54573
START_RANGE client27b socket856 630 54573-54636
START_RANGE client27a socket857 640 54636-54700
START_RANGE client27b socket858 640 54700-54764
START_RANGE client27a socket859 640 54764-54828
START_RANGE client27b socket860 640 54828-54892
START_RANGE client27a socket861 640 54892-54956
START_RANGE client27b socket862 630 54956-55019
START_RANGE client27a socket863 640 55019-55083
START_RANGE client27b socket864 640 55083-55147

START_RANGE client28a socket865 640 55147-55211
START_RANGE client28b socket866 640 55211-55275
START_RANGE client28a socket867 630 55275-55338
START_RANGE client28b socket868 640 55338-55402
START_RANGE client28a socket869 640 55402-55466
START_RANGE client28b socket870 640 55466-55530
START_RANGE client28a socket871 640 55530-55594
START_RANGE client28b socket872 640 55594-55658
START_RANGE client28a socket873 630 55658-55721
START_RANGE client28b socket874 640 55721-55785
START_RANGE client28a socket875 640 55785-55849
START_RANGE client28b socket876 640 55849-55913
START_RANGE client28a socket877 640 55913-55977
START_RANGE client28b socket878 640 55977-56041
START_RANGE client28a socket879 630 56041-56104
START_RANGE client28b socket880 640 56104-56168

START_RANGE client28a socket881 640 56168-56232
START_RANGE client28b socket882 640 56232-56296
START_RANGE client28a socket883 640 56296-56360
START_RANGE client28b socket884 640 56360-56424
START_RANGE client28a socket885 630 56424-56487
START_RANGE client28b socket886 640 56487-56551
START_RANGE client28a socket887 640 56551-56615
START_RANGE client28b socket888 640 56615-56679
START_RANGE client28a socket889 640 56679-56743
START_RANGE client28b socket890 640 56743-56807
START_RANGE client28a socket891 630 56807-56870
START_RANGE client28b socket892 640 56870-56934
START_RANGE client28a socket893 640 56934-56998
START_RANGE client28b socket894 640 56998-57062
START_RANGE client28a socket895 640 57062-57126
START_RANGE client28b socket896 640 57126-57190

#elif MASTER_NUM8

START_RANGE client29a socket897 630 57190-57253
START_RANGE client29b socket898 640 57253-57317
START_RANGE client29a socket899 640 57317-57381
START_RANGE client29b socket900 640 57381-57445
START_RANGE client29a socket901 640 57445-57509
START_RANGE client29b socket902 630 57509-57572
START_RANGE client29a socket903 640 57572-57636
START_RANGE client29b socket904 640 57636-57700
START_RANGE client29a socket905 640 57700-57764
START_RANGE client29b socket906 640 57764-57828
START_RANGE client29a socket907 640 57828-57892
START_RANGE client29b socket908 630 57892-57955
START_RANGE client29a socket909 640 57955-58019
START_RANGE client29b socket910 640 58019-58083
START_RANGE client29a socket911 640 58083-58147
START_RANGE client29b socket912 640 58147-58211

START_RANGE client29a socket913 640 58211-58275
START_RANGE client29b socket914 630 58275-58338
START_RANGE client29a socket915 640 58338-58402
START_RANGE client29b socket916 640 58402-58466
START_RANGE client29a socket917 640 58466-58530
START_RANGE client29b socket918 640 58530-58594
START_RANGE client29a socket919 640 58594-58658
START_RANGE client29b socket920 630 58658-58721
START_RANGE client29a socket921 640 58721-58785
START_RANGE client29b socket922 640 58785-58849
START_RANGE client29a socket923 640 58849-58913
START_RANGE client29b socket924 640 58913-58977
START_RANGE client29a socket925 640 58977-59041
START_RANGE client29b socket926 630 59041-59104
START_RANGE client29a socket927 640 59104-59168
START_RANGE client29b socket928 640 59168-59232

START_RANGE client30a socket929 640 59232-59296
START_RANGE client30b socket930 640 59296-59360
START_RANGE client30a socket931 630 59360-59423
START_RANGE client30b socket932 640 59423-59487
START_RANGE client30a socket933 640 59487-59551
START_RANGE client30b socket934 640 59551-59615
START_RANGE client30a socket935 640 59615-59679
START_RANGE client30b socket936 640 59679-59743
START_RANGE client30a socket937 630 59743-59806
START_RANGE client30b socket938 640 59806-59870
START_RANGE client30a socket939 640 59870-59934
START_RANGE client30b socket940 640 59934-59998
START_RANGE client30a socket941 640 59998-60062
START_RANGE client30b socket942 640 60062-60126
START_RANGE client30a socket943 630 60126-60189
START_RANGE client30b socket944 640 60189-60253

START_RANGE client30a socket945 640 60253-60317
START_RANGE client30b socket946 640 60317-60381
START_RANGE client30a socket947 640 60381-60445
START_RANGE client30b socket948 640 60445-60509
START_RANGE client30a socket949 630 60509-60572
START_RANGE client30b socket950 640 60572-60636
START_RANGE client30a socket951 640 60636-60700
START_RANGE client30b socket952 640 60700-60764
START_RANGE client30a socket953 640 60764-60828
START_RANGE client30b socket954 640 60828-60892
START_RANGE client30a socket955 630 60892-60955
START_RANGE client30b socket956 640 60955-61019
START_RANGE client30a socket957 640 61019-61083
START_RANGE client30b socket958 640 61083-61147
START_RANGE client30a socket959 640 61147-61211
START_RANGE client30b socket960 640 61211-61275

START_RANGE client31a socket961 630 61275-61338
START_RANGE client31b socket962 640 61338-61402
START_RANGE client31a socket963 640 61402-61466
START_RANGE client31b socket964 640 61466-61530
START_RANGE client31a socket965 640 61530-61594
START_RANGE client31b socket966 630 61594-61657
START_RANGE client31a socket967 640 61657-61721
START_RANGE client31b socket968 640 61721-61785
START_RANGE client31a socket969 640 61785-61849
START_RANGE client31b socket970 640 61849-61913
START_RANGE client31a socket971 640 61913-61977
START_RANGE client31b socket972 630 61977-62040
START_RANGE client31a socket973 640 62040-62104
START_RANGE client31b socket974 640 62104-62168
START_RANGE client31a socket975 640 62168-62232
START_RANGE client31b socket976 640 62232-62296

START_RANGE client31a socket977 640 62296-62360
START_RANGE client31b socket978 630 62360-62423
START_RANGE client31a socket979 640 62423-62487
START_RANGE client31b socket980 640 62487-62551

START_RANGE client31a socket981 640 62551-62615
START_RANGE client31b socket982 640 62615-62679
START_RANGE client31a socket983 640 62679-62743
START_RANGE client31b socket984 630 62743-62806
START_RANGE client31a socket985 640 62806-62870
START_RANGE client31b socket986 640 62870-62934
START_RANGE client31a socket987 640 62934-62998
START_RANGE client31b socket988 640 62998-63062
START_RANGE client31a socket989 640 63062-63126
START_RANGE client31b socket990 630 63126-63189
START_RANGE client31a socket991 640 63189-63253
START_RANGE client31b socket992 640 63253-63317

START_RANGE client32a socket993 640 63317-63381
START_RANGE client32b socket994 640 63381-63445
START_RANGE client32a socket995 630 63445-63508
START_RANGE client32b socket996 640 63508-63572
START_RANGE client32a socket997 640 63572-63636
START_RANGE client32b socket998 640 63636-63700
START_RANGE client32a socket999 640 63700-63764
START_RANGE client32b socket1000 640 63764-63828
START_RANGE client32a socket1001 630 63828-63891
START_RANGE client32b socket1002 640 63891-63955
START_RANGE client32a socket1003 640 63955-64019
START_RANGE client32b socket1004 640 64019-64083
START_RANGE client32a socket1005 640 64083-64147
START_RANGE client32b socket1006 640 64147-64211
START_RANGE client32a socket1007 630 64211-64274
START_RANGE client32b socket1008 640 64274-64338

START_RANGE client32a socket1009 640 64338-64402
START_RANGE client32b socket1010 640 64402-64466
START_RANGE client32a socket1011 640 64466-64530
START_RANGE client32b socket1012 640 64530-64594
START_RANGE client32a socket1013 630 64594-64657
START_RANGE client32b socket1014 640 64657-64721
START_RANGE client32a socket1015 640 64721-64785
START_RANGE client32b socket1016 640 64785-64849
START_RANGE client32a socket1017 640 64849-64913
START_RANGE client32b socket1018 640 64913-64977
START_RANGE client32a socket1019 630 64977-65040
START_RANGE client32b socket1020 640 65040-65104
START_RANGE client32a socket1021 640 65104-65168
START_RANGE client32b socket1022 640 65168-65232
START_RANGE client32a socket1023 640 65232-65296
START_RANGE client32b socket1024 640 65296-65360

#elif MASTER_NUM9

START_RANGE client33a socket1025 630 65360-65423
START_RANGE client33b socket1026 640 65423-65487
START_RANGE client33a socket1027 640 65487-65551
START_RANGE client33b socket1028 640 65551-65615
START_RANGE client33a socket1029 640 65615-65679
START_RANGE client33b socket1030 630 65679-65742
START_RANGE client33a socket1031 640 65742-65806
START_RANGE client33b socket1032 640 65806-65870
START_RANGE client33a socket1033 640 65870-65934
START_RANGE client33b socket1034 640 65934-65998
START_RANGE client33a socket1035 640 65998-66062
START_RANGE client33b socket1036 630 66062-66125
START_RANGE client33a socket1037 640 66125-66189
START_RANGE client33b socket1038 640 66189-66253
START_RANGE client33a socket1039 640 66253-66317
START_RANGE client33b socket1040 640 66317-66381

START_RANGE client33a socket1041 640 66381-66445
START_RANGE client33b socket1042 630 66445-66508
START_RANGE client33a socket1043 640 66508-66572
START_RANGE client33b socket1044 640 66572-66636
START_RANGE client33a socket1045 640 66636-66700
START_RANGE client33b socket1046 640 66700-66764
START_RANGE client33a socket1047 640 66764-66828

START_RANGE client33b socket1048 630 66828-66891
START_RANGE client33a socket1049 640 66891-66955
START_RANGE client33b socket1050 640 66955-67019
START_RANGE client33a socket1051 640 67019-67083
START_RANGE client33b socket1052 640 67083-67147
START_RANGE client33a socket1053 640 67147-67211
START_RANGE client33b socket1054 630 67211-67274
START_RANGE client33a socket1055 640 67274-67338
START_RANGE client33b socket1056 640 67338-67402

START_RANGE client34a socket1057 640 67402-67466
START_RANGE client34b socket1058 640 67466-67530
START_RANGE client34a socket1059 630 67530-67593
START_RANGE client34b socket1060 640 67593-67657
START_RANGE client34a socket1061 640 67657-67721
START_RANGE client34b socket1062 640 67721-67785
START_RANGE client34a socket1063 640 67785-67849
START_RANGE client34b socket1064 640 67849-67913
START_RANGE client34a socket1065 630 67913-67976
START_RANGE client34b socket1066 640 67976-68040
START_RANGE client34a socket1067 640 68040-68104
START_RANGE client34b socket1068 640 68104-68168
START_RANGE client34a socket1069 640 68168-68232
START_RANGE client34b socket1070 640 68232-68296
START_RANGE client34a socket1071 630 68296-68359
START_RANGE client34b socket1072 640 68359-68423

START_RANGE client34a socket1073 640 68423-68487
START_RANGE client34b socket1074 640 68487-68551
START_RANGE client34a socket1075 640 68551-68615
START_RANGE client34b socket1076 640 68615-68679
START_RANGE client34a socket1077 630 68679-68742
START_RANGE client34b socket1078 640 68742-68806
START_RANGE client34a socket1079 640 68806-68870
START_RANGE client34b socket1080 640 68870-68934
START_RANGE client34a socket1081 640 68934-68998
START_RANGE client34b socket1082 640 68998-69062
START_RANGE client34a socket1083 630 69062-69125
START_RANGE client34b socket1084 640 69125-69189
START_RANGE client34a socket1085 640 69189-69253
START_RANGE client34b socket1086 640 69253-69317
START_RANGE client34a socket1087 640 69317-69381
START_RANGE client34b socket1088 640 69381-69445

START_RANGE client35a socket1089 630 69445-69508
START_RANGE client35b socket1090 640 69508-69572
START_RANGE client35a socket1091 640 69572-69636
START_RANGE client35b socket1092 640 69636-69700
START_RANGE client35a socket1093 640 69700-69764
START_RANGE client35b socket1094 630 69764-69827
START_RANGE client35a socket1095 640 69827-69891
START_RANGE client35b socket1096 640 69891-69955
START_RANGE client35a socket1097 640 69955-70019
START_RANGE client35b socket1098 640 70019-70083
START_RANGE client35a socket1099 640 70083-70147
START_RANGE client35b socket1100 630 70147-70210
START_RANGE client35a socket1101 640 70210-70274
START_RANGE client35b socket1102 640 70274-70338
START_RANGE client35a socket1103 640 70338-70402
START_RANGE client35b socket1104 640 70402-70466

START_RANGE client35a socket1105 640 70466-70530
START_RANGE client35b socket1106 630 70530-70593
START_RANGE client35a socket1107 640 70593-70657
START_RANGE client35b socket1108 640 70657-70721
START_RANGE client35a socket1109 640 70721-70785
START_RANGE client35b socket1110 640 70785-70849
START_RANGE client35a socket1111 640 70849-70913
START_RANGE client35b socket1112 630 70913-70976
START_RANGE client35a socket1113 640 70976-71040
START_RANGE client35b socket1114 640 71040-71104
START_RANGE client35a socket1115 640 71104-71168

START_RANGE client35b socket1116 640 71168-71232
START_RANGE client35a socket1117 640 71232-71296
START_RANGE client35b socket1118 630 71296-71359
START_RANGE client35a socket1119 640 71359-71423
START_RANGE client35b socket1120 640 71423-71487

START_RANGE client36a socket1121 640 71487-71551
START_RANGE client36b socket1122 640 71551-71615
START_RANGE client36a socket1123 630 71615-71678
START_RANGE client36b socket1124 640 71678-71742
START_RANGE client36a socket1125 640 71742-71806
START_RANGE client36b socket1126 640 71806-71870
START_RANGE client36a socket1127 640 71870-71934
START_RANGE client36b socket1128 640 71934-71998
START_RANGE client36a socket1129 630 71998-72061
START_RANGE client36b socket1130 640 72061-72125
START_RANGE client36a socket1131 640 72125-72189
START_RANGE client36b socket1132 640 72189-72253
START_RANGE client36a socket1133 640 72253-72317
START_RANGE client36b socket1134 640 72317-72381
START_RANGE client36a socket1135 630 72381-72444
START_RANGE client36b socket1136 640 72444-72508

START_RANGE client36a socket1137 640 72508-72572
START_RANGE client36b socket1138 640 72572-72636
START_RANGE client36a socket1139 640 72636-72700
START_RANGE client36b socket1140 640 72700-72764
START_RANGE client36a socket1141 630 72764-72827
START_RANGE client36b socket1142 640 72827-72891
START_RANGE client36a socket1143 640 72891-72955
START_RANGE client36b socket1144 640 72955-73019
START_RANGE client36a socket1145 640 73019-73083
START_RANGE client36b socket1146 640 73083-73147
START_RANGE client36a socket1147 630 73147-73210
START_RANGE client36b socket1148 640 73210-73274
START_RANGE client36a socket1149 640 73274-73338
START_RANGE client36b socket1150 640 73338-73402
START_RANGE client36a socket1151 640 73402-73466
START_RANGE client36b socket1152 640 73466-73530

#elif MASTER_NUM10

START_RANGE client37a socket1153 630 73530-73593
START_RANGE client37b socket1154 640 73593-73657
START_RANGE client37a socket1155 640 73657-73721
START_RANGE client37b socket1156 640 73721-73785
START_RANGE client37a socket1157 640 73785-73849
START_RANGE client37b socket1158 630 73849-73912
START_RANGE client37a socket1159 640 73912-73976
START_RANGE client37b socket1160 640 73976-74040
START_RANGE client37a socket1161 640 74040-74104
START_RANGE client37b socket1162 640 74104-74168
START_RANGE client37a socket1163 640 74168-74232
START_RANGE client37b socket1164 630 74232-74295
START_RANGE client37a socket1165 640 74295-74359
START_RANGE client37b socket1166 640 74359-74423
START_RANGE client37a socket1167 640 74423-74487
START_RANGE client37b socket1168 640 74487-74551

START_RANGE client37a socket1169 640 74551-74615
START_RANGE client37b socket1170 630 74615-74678
START_RANGE client37a socket1171 640 74678-74742
START_RANGE client37b socket1172 640 74742-74806
START_RANGE client37a socket1173 640 74806-74870
START_RANGE client37b socket1174 640 74870-74934
START_RANGE client37a socket1175 640 74934-74998
START_RANGE client37b socket1176 630 74998-75061
START_RANGE client37a socket1177 640 75061-75125
START_RANGE client37b socket1178 640 75125-75189
START_RANGE client37a socket1179 640 75189-75253
START_RANGE client37b socket1180 640 75253-75317
START_RANGE client37a socket1181 640 75317-75381
START_RANGE client37b socket1182 630 75381-75444

START_RANGE client37a socket1183 640 75444-75508
START_RANGE client37b socket1184 640 75508-75572

START_RANGE client38a socket1185 640 75572-75636
START_RANGE client38b socket1186 640 75636-75700
START_RANGE client38a socket1187 630 75700-75763
START_RANGE client38b socket1188 640 75763-75827
START_RANGE client38a socket1189 640 75827-75891
START_RANGE client38b socket1190 640 75891-75955
START_RANGE client38a socket1191 640 75955-76019
START_RANGE client38b socket1192 640 76019-76083
START_RANGE client38a socket1193 630 76083-76146
START_RANGE client38b socket1194 640 76146-76210
START_RANGE client38a socket1195 640 76210-76274
START_RANGE client38b socket1196 640 76274-76338
START_RANGE client38a socket1197 640 76338-76402
START_RANGE client38b socket1198 640 76402-76466
START_RANGE client38a socket1199 630 76466-76529
START_RANGE client38b socket1200 640 76529-76593

START_RANGE client38a socket1201 640 76593-76657
START_RANGE client38b socket1202 640 76657-76721
START_RANGE client38a socket1203 640 76721-76785
START_RANGE client38b socket1204 640 76785-76849
START_RANGE client38a socket1205 630 76849-76912
START_RANGE client38b socket1206 640 76912-76976
START_RANGE client38a socket1207 640 76976-77040
START_RANGE client38b socket1208 640 77040-77104
START_RANGE client38a socket1209 640 77104-77168
START_RANGE client38b socket1210 640 77168-77232
START_RANGE client38a socket1211 630 77232-77295
START_RANGE client38b socket1212 640 77295-77359
START_RANGE client38a socket1213 640 77359-77423
START_RANGE client38b socket1214 640 77423-77487
START_RANGE client38a socket1215 640 77487-77551
START_RANGE client38b socket1216 640 77551-77615

START_RANGE client39a socket1217 630 77615-77678
START_RANGE client39b socket1218 640 77678-77742
START_RANGE client39a socket1219 640 77742-77806
START_RANGE client39b socket1220 640 77806-77870
START_RANGE client39a socket1221 640 77870-77934
START_RANGE client39b socket1222 630 77934-77997
START_RANGE client39a socket1223 640 77997-78061
START_RANGE client39b socket1224 640 78061-78125
START_RANGE client39a socket1225 640 78125-78189
START_RANGE client39b socket1226 640 78189-78253
START_RANGE client39a socket1227 640 78253-78317
START_RANGE client39b socket1228 630 78317-78380
START_RANGE client39a socket1229 640 78380-78444
START_RANGE client39b socket1230 640 78444-78508
START_RANGE client39a socket1231 640 78508-78572
START_RANGE client39b socket1232 640 78572-78636

START_RANGE client39a socket1233 640 78636-78700
START_RANGE client39b socket1234 630 78700-78763
START_RANGE client39a socket1235 640 78763-78827
START_RANGE client39b socket1236 640 78827-78891
START_RANGE client39a socket1237 640 78891-78955
START_RANGE client39b socket1238 640 78955-79019
START_RANGE client39a socket1239 640 79019-79083
START_RANGE client39b socket1240 630 79083-79146
START_RANGE client39a socket1241 640 79146-79210
START_RANGE client39b socket1242 640 79210-79274
START_RANGE client39a socket1243 640 79274-79338
START_RANGE client39b socket1244 640 79338-79402
START_RANGE client39a socket1245 640 79402-79466
START_RANGE client39b socket1246 630 79466-79529
START_RANGE client39a socket1247 640 79529-79593
START_RANGE client39b socket1248 640 79593-79657

START_RANGE client40a socket1249 640 79657-79721

START_RANGE client40b socket1250 640 79721-79785
START_RANGE client40a socket1251 630 79785-79848
START_RANGE client40b socket1252 640 79848-79912
START_RANGE client40a socket1253 640 79912-79976
START_RANGE client40b socket1254 640 79976-80040
START_RANGE client40a socket1255 640 80040-80104
START_RANGE client40b socket1256 640 80104-80168
START_RANGE client40a socket1257 630 80168-80231
START_RANGE client40b socket1258 640 80231-80295
START_RANGE client40a socket1259 640 80295-80359
START_RANGE client40b socket1260 640 80359-80423
START_RANGE client40a socket1261 640 80423-80487
START_RANGE client40b socket1262 640 80487-80551
START_RANGE client40a socket1263 630 80551-80614
START_RANGE client40b socket1264 640 80614-80678

START_RANGE client40a socket1265 640 80678-80742
START_RANGE client40b socket1266 640 80742-80806
START_RANGE client40a socket1267 640 80806-80870
START_RANGE client40b socket1268 640 80870-80934
START_RANGE client40a socket1269 630 80934-80997
START_RANGE client40b socket1270 640 80997-81061
START_RANGE client40a socket1271 640 81061-81125
START_RANGE client40b socket1272 640 81125-81189
START_RANGE client40a socket1273 640 81189-81253
START_RANGE client40b socket1274 640 81253-81317
START_RANGE client40a socket1275 630 81317-81380
START_RANGE client40b socket1276 640 81380-81444
START_RANGE client40a socket1277 640 81444-81508
START_RANGE client40b socket1278 640 81508-81572
START_RANGE client40a socket1279 640 81572-81636
START_RANGE client40b socket1280 640 81636-81700

```
#elif MASTER_NUM11
#endif
/*-----*/
#define TES_FLAG_TRACE 0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x000000200
#define TES_FLAG_LOCAL_LOG 0x000000400
#define TES_FLAG_LOCAL_TRACE 0x000000800
#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
////////////////////////////////////

#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
////////////////////////////////////

#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=%d"

#define PAYMENT_RESULTS_CID_URL "/tpcc/tpcc.html?00=pymt&01=%d&03=%d&04=%d&05=%d&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="

#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 strtocmp(char *s1, char *s2);
//extern "C" int strncasecmp(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 %6s %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 */
}
iprint(IPRINT_INFO, "Login Timeout = %s\n",
mstoa_withfrac(shmglobal->login_timeout, 0));
iprint(IPRINT_INFO, "Keystroke Sleep = %s\n",
mstoa_withfrac(shmglobal->keystroke_sleep*1000, 0));
iprint(IPRINT_INFO, "Keystroke Packet Size= %d\n", shmglobal-
>keystroke_packet_size);
if (num_warehouses >= 0) {
iprint(IPRINT_INFO, "Fixed Warehouses to = %d\n", num_warehouses);
}

if (!(ptr = get_variable("NEWORDER"))) {
iprint_error ("Error. NEWORDER variable not found\n");
exit (1);
}
if (parse_array(ptr, 3, dbuffer)!=3) {
iprint_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)) {
iprint(__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 WHSEARRAYYDBG
    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 (!strcmp (ptr, "pause")) {
        shmglobal->test_state = 1;
    } else if (!strcmp (ptr, "warmup")) {
        shmglobal->test_state = 2;
    } else if (!strcmp (ptr, "notest")) {
        shmglobal->test_state = 0;
    } else if (!strcmp (ptr, "login_max_load?")) {
        fprintf (IPRINT_WARNING, "Current LOGIN_MAX_LOAD = %d\n",
            login_max_load);
    } else if (!strcmp (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 (!strcmp (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 (!strcmp (ptr, Status_Names[i], len)) {
                    status_needs_refresh = found = 1;
                    current_status = i;
                    return RTE_OK;
                }
            }
        }
    }
}

```

```

    }
    fprintf (IPRINT_WARNING, "Unknown type to display: %s\n", ptr);
}
} else {
    fprintf (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 KEYWAIT_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 = "tperrno";
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);
#ifdef 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_RESULTS_TITLE_L
            EN);
            break;
        case '2':
            rc =
            expect_html(PYMT_RESULTS_TITLE,timeout,PYMT_RESULTS_TITLE_L
            EN);
            break;
        case '3':
            rc =
            expect_html(ORDS_RESULTS_TITLE,timeout,ORDS_RESULTS_TITLE_L
            EN);
            break;
        case '4':
            rc =
            expect_html(DLVY_RESULTS_TITLE,timeout,DLVY_RESULTS_TITLE_L
            EN);
            break;
        case '5':
            rc =
            expect_html(STOK_RESULTS_TITLE,timeout,STOK_RESULTS_TITLE_LE
            N);
            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];
    charform_buffer[256];
    charresults_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

```

```

printf(dlvy_url,DELIVERY_RESULTS_URL,terminal,delivery_new.carrier
);
printf(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];
charform_buffer[512];

char nord_results_url[2048];
charresults_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); // number of items to
order (5-15)
neword_new.olremote = 0; // find total number of
remote order-lines

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
printf(nord_form_url,NEW_ORDER_FORM_URL,terminal);

//create get form request
printf(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);
}

printf(nord_results_url,NEW_ORDER_RESULTS_URL,terminal,
neword_new.did,neword_new.cid,
itemString);

//create get results request
printf(results_buffer,GET_REQUEST,nord_results_url,host);

// 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");
    newword.oid = -1;
}
else
{
    newword.oid = atoi(err_ptr+14);
    // iprint(IPRINT_ERROR,"New order order id:%d\n",newword.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_URL,terminal,ordstat_new.did,
ordstat_new.clast);
else

    sprintf(ords_url,ORDER_STATUS_RESULTS_CID_URL,terminal,ordstat_new.did,
ordstat_new.cid);

    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;

```



```

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);
}

/*
switch(tmp) {
case Delivery: return Payment(host,terminal);
case OrderStatus: return Payment(host,terminal);
case Payment: return Payment(host,terminal);
case StockLevel: return Payment(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_ctux_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=\"", 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 ITEMC 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 cdid;
    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 successfully */
};

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: Pricing Info

Address <http://www.microsoft.com/windows2000/server/howtobuy/pricing/default.asp>

All Products | Support | Search | Microsoft.com Guide



Windows Home | Windows 2000 Home | Windows 2000 Worldwide

Search for: Go

Advanced Search

- Windows 2000
- Product Information
- How to Buy
- Technical Resources
- Downloads
- Support
- Technology Centers
- Partners
- Windows Server Community
- Windows Family
- Windows Update

[Windows 2000](#)

Windows 2000 Pricing and Licensing

This page lists pricing and licensing details for U.S. and Canadian editions of Windows® 2000 Server.



[Professional](#)
[Server](#)
[Advanced Server](#)
[Datacenter Server](#)

ON THIS PAGE

- [Estimated Retail Pricing](#)
- [Introduction to Windows 2000 Licensing](#)
- [Find Pricing and Upgrading Information](#)
- [Volume Licensing](#)
- [For More Information](#)

Estimated Retail Pricing

Microsoft has designed separate client and server licensing components to allow complete scalability of your client/server solution. In addition, Microsoft offers several flexible, cost-effective options for licensing. This page will help you find the Windows 2000 Server pricing and licensing strategy that is most cost-effective for your organization. For additional details and purchasing information, visit the Microsoft Product Catalog pages for [Windows 2000 Server](#) and [Windows 2000 Advanced Server](#). Or, order the product of your choice from the table below.

Product Offering	Order Now	Price	Description
Windows 2000 Server (With 10 Client Access Licenses)		\$1199 US	Standard product plus 10 CALs
Windows 2000 Server (With 5 Client Access Licenses)		\$999 US	Standard product plus 5 CALs

Search for

Microsoft® Visual Studio® Developer Center

Rapidly build applications for Microsoft Windows®, the Web, and mobile devices



[MSDN Home](#) > [Visual Studio](#) > [How to Buy](#) > [Look Up Prices](#)

- Advanced Search
- Visual Studio Home
 - Product Information
 - How to Buy
 - Registration
 - Using Visual Studio .NET
 - Downloads
 - Support
 - Community
 - Partners
 - Visual Studio Whitebox Information
 - Previous Versions
 - Products & Technologies

Look Up Prices

For a feature comparison of the different editions of Visual Studio .NET, see the [Choose Your Edition](#) feature comparison page.

Licensed users of Visual Studio .NET version 2002 Professional, Enterprise Developer, or Enterprise Architect editions, who have active Microsoft Volume Licensing agreements should have already received Visual Studio .NET 2003 directly from Microsoft. For details, visit the [Visual Studio .NET 2003 upgrade page](#).

Full and Upgrade Pricing

These are estimated retail prices in United States dollars—reseller pricing may vary. To find the Visual Studio .NET Web site for your country/region, visit the [Microsoft Worldwide](#) Web site. You can also order Visual Studio .NET Professional, Enterprise Developer, and Enterprise Architect as part of an MSDN Subscription, which provides priority access to Visual Studio .NET and the technologies you need to build and fully test XML Web services and applications.

Visual Studio .NET 2003 Professional

Related Links

- [How to Buy Visual Studio .NET 2003](#)
- [Visual Studio .NET 2003 Purchasing Options](#)
- [Visual Basic .NET Standard](#)
- [Visual C# .NET Standard](#)
- [Visual C++ .NET Standard](#)



Type	Estimated Price	Upgrade Information	Rebate Information
Full Packaged Product	\$1,079 US	Not Applicable	Not Applicable
Version Upgrade	\$549 US	<p>Customers who qualify for this upgrade price include licensed users of the following Microsoft and competitor products:</p> <ul style="list-style-type: none"> • Visual Studio .NET 2002 Professional Edition • Visual Studio .NET 2002 Enterprise Developer • Visual Studio .NET 2002 Enterprise Architect • Visual Studio 97 Professional Edition • Visual Studio 97 Enterprise Edition • Visual Studio 6.0 Professional Edition • Visual Studio 6.0 Enterprise Edition • Visual Basic® 5.0 (or later) Professional Edition • Visual Basic 5.0 (or later) Enterprise Edition • Visual C++® 5.0 (or later) Professional Edition • Visual C++ 5.0 (or later) Enterprise Edition, • Visual InterDev® 1.0 (or later) • Visual J++® 1.1 (or later) Professional Edition • Visual FoxPro® 5.0 (or later) Professional Edition • Visual SourceSafe® 5.0 (or later) 	Not Applicable

NETGEAR FSM750S Managed Stackable Switch with Gigabit ports - switch - 48 ports



[At-A-Glance](#) | [Overview](#) | [Tech Specs](#) | [Rebate](#)

Mfr Part #: FSM750SNA

- Layer 2 management (SNMP, RMON, VLAN, QoS, port trunking, and more)
- 48 10/100 front ports
- 2 Gigabit Ethernet front ports
- 2 rear stacking ports to stack 6 units high

NETGEAR
Everybody's connecting.™

NETGEAR's non-blocking FSM750S is a Layer 2 managed switch that's ideal for anyone new to managed switching. With 48 10/100 ports, 2 rear stacking ports, and 2 gigabit ports -- use either built-in Copper Gigabit or GBIC slots -- small businesses can keep pace with their increasing workflow while accommodating up to 150 network users. The versatile FSM750S offers two interface options: an easy-to-follow, browser-based interface and a command line interface. Management features inc ... [\[continue\]](#)

- [Search for Similar Products](#)
- [Email This Page](#)

Add to Order

USD **\$758.73**

[195 In Stock.](#)

Qty:

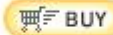
Add to Order

ADAPTEC QUARTET66 (ANA-64044LV)



For 10/100 Mbps Ethernet LANs ANA-64044LV
4-Port, 64-bit/66 MHz PCI network interface card
for Fast Ethernet environments

SRP (US) \$550.00 Part Number: 1932500



View product details in:

[English](#)

[\(UK\), Deutsch, Français, Italiano, Español](#)

The Adaptec® ANA-64044LV is a 66MHz four port NIC offering superior data transfer performance through port aggregation. Four ports and the included Duralink64 Failover software work to ensure network availability. This NIC is already iSCSI-Ready for the next generation of IP Storage SANs.

Highlights

- Peak performance with 64 bit / 66MHz bus design and port aggregation capabilities (maximum 800Mbps at full duplex)
- Dynamic Load Balancing between 4-ports for consistent, optimal performance
- Failover security provided with failover software and redundant port availability
- iSCSI Ready
- [Complete List of Supported Operating Systems](#)

For further product inquiries, please contact Adaptec Sales:

U.S. & Canada: 1.800.442.7274 | email (English only): Sales@Adaptec.com

PRODUCT DETAILS

- > [Product Specifications](#)
- > [DuraLAN Fast Ethernet Network Interface Card Family Datasheet \(PDF, 571 kb\)](#)
- > [Product Family Comparison](#)

PRODUCT RESOURCES

- > [Product Support](#)
- > [Product Downloads](#)
- > [Set-up & Installation](#)
- > [A.S.K. Knowledgebase Articles](#)
- > [Register My Product](#)

[Site Index](#) | [Legal](#) | [Privacy Policy](#) | [Comments/Questions](#) | [Terms of Use](#) | ©1995 - 2004 Adaptec, Inc.

Your Privacy Matters at Adaptec!